[*MT Summit III*, Washington, DC, USA, July 1991]

KIELIKONE Machine Translation Workstation

H. Jäppinen, L. Kulikov, A. Ylä-Rotiala
SITRA Foundation, KIELIKONE-project
P.O. Box 329,
00121 Helsinki,
Finland

Abstract

All human languages are open and complex communication systems. No machine translation system will ever be able to automatically translate all possible sentences from one language to another in high quality. One way to combat complexity and openness of language translation is to decompose the task into well-defined sequential subtasks and solve each using declarative, modular rules. This paper describes such an MT system. A language-independent MT Machine has been designed for the transformation of linguistic trees in a general fashion. A full MT system is composed of a sequence of instances of that machine. Finnish-English implementation is discussed.

1 Introduction

The great majority of Finns speak a language which differs radically from main Indo-European languages. Finnish is highly inflectional and words have potentially thousands of distinct forms. Word forms carry syntactic information in their suffixes and therefore word order is relatively free in Finnish sentences. Because Finnish is syntactically so different from most other Western languages, Finns face a higher language barrier than other Western Europeans do. Increasing foreign trade has forced major Finnish companies to systematically look for ways of making language translation more productive. Machine translation would of course seem to provide an ideal solution, but in practice both the state-of-the-art of MT research and the lack of computational models of Finnish have so far discouraged the companies in their attempts to apply MT software to alleviate the translation load.

SITRA Foundation in Finland is a public fund which allocates money for projects of notable national importance. In 1982 SITRA established the KIELIKONE project for the purpose of designing computational models of the Finnish language. The short term goals were to obtain concrete language technology products; the simultaneous long term goal was to build an infrastructure for MT research. During its period of activity so far the project has designed, implemented, and introduced to the market various software products for the Finnish language: a morphological analyzer (Jäppinen and Ylilammi, 1986) and spelling checkers based on that model, a morphological synthesizer (Lassila, 1988), a hyphenation algorithm, and dependency parsers (Nelimarkka et al., 1984; Jäppinen et al., 1986; Valkonen et al., 1987; Lassila, 1989). Also, a synonym dictionary for Finnish has been produced both in book (Jäppinen, 1989) and electronic form.

As more direct steps toward MT, the project first developed an electronic bilingual Finnish-English dictionary. Later on, upon the request of a foreign customer, the project designed and implemented an MT workstation for a syntactically and semantically constrained sublanguage {Kulikov and Jäppinen, 1989; Takala et al., 1990).

In 1986 it was decided that the project should concentrate on full-scale MT research in cooperation with two major Finnish companies. Telenokia OY exports telecommunication equipment, and all of their products require extensive technical documentation. English is their most important foreign language; this company is our pilot customer for the Finnish-English system. Finnair OY is the national Finnish air carrier company. Their main problem is the translation of voluminous maintenance manuals from English into Finnish. This company is the pilot customer for our English-Finnish system.

The focus of our MT research has been the design of MT Workstations. By this term we mean personal computing systems which produce good quality raw translations and support post-editing with a user-friendly linguistic editor. To promote wide applicability the system architecture is designed to be maximally general (language independent), and the part which holds language-dependent definitions is declarative. These principles have been realized in an MT Machine, which holds the algorithmic part of any given MT Workstation implementation. The MT Machine is totally language independent — it is not biased towards Finnish — and its execution is controlled by a declarative rule base.

As of this writing, we have fully implemented and tested the MT Machine (in C under UNIX). A Finnish-English Workstation has also been implemented and we are presently testing and tuning the system using real data.

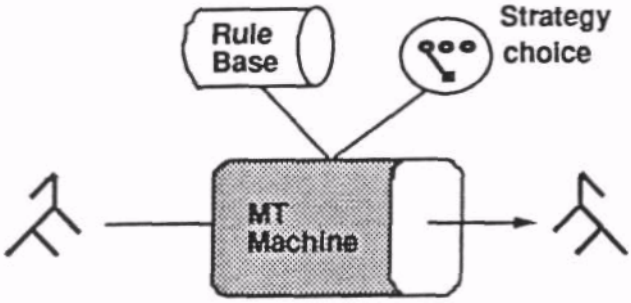

Figure 1: The MT Machine

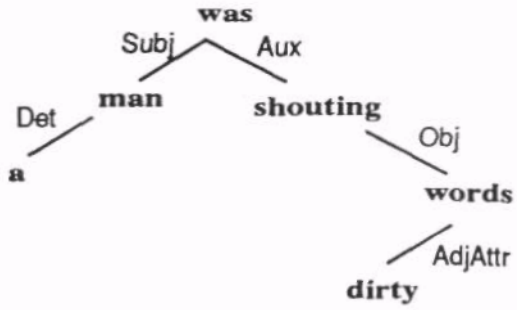

Figure 2: A dependency tree

2 The MT Machine

The MT Machine is a general tree-manipulation system with several built-in inference strategies. When a user applies the machine he/she writes a rule base to control the execution of the machine and chooses the appropriate inference strategy. The machine takes well-defined linguistic trees as input and produces as output trees which represent meaning-preserving transformations of the input trees (Fig. 1).

We will not discuss either the rule syntax or the inference strategies here. As for the linguistic trees, they are general feature trees (F-trees); the nodes of trees are represented by feature vectors.

Although the MT Machine is general, i.e. language independent, it does impose restrictions on what kinds of transformations are possible. The tree topology rules out, for instance, graph manipulation. The chosen rule syntax and the implemented inference strategies impose limitations of their own, but it is our belief that these restrictions are linguistically well-founded and do not constrain translations. The experience gathered with the Finnish-English Workstation system so far supports this conjecture.

It is important to notice in a positive sense how the MT Machine enables homogeneous processing. The data flow is in the form of F-trees throughout the process and descriptions of transformations are always rule bases (even lexicons are rule bases in our implementations), Processing corresponds to a monotonous application of F-tree transformations (Fig. 3). Homogeneity has many advantages; it means structural simplicity and thus advances clarity and maintainability,

3 Linguistic Commitments

The MT Machine itself is not confined to the use of any specific linguistic theory. We have committed ourselves in our implementations to dependency theory as the model of sentence structure. We have studied dependency theory over the years and implemented parsers of Finnish based on it (Nelimarkka et al., 1984; Jäppinen et al,, 1986; Valkonen et al., 1987; Lassila, 1989). Dependency theory, we have argued, describes the sentence structure of so-called free-word-order languages better than constituent theories do.

Dependency trees do not explicitly show the constituent structure of a sentence. Instead, they exhibit the binary head-modifier relations between the words. The result of a parsing process is hence a tree whose nodes represent the words (more specifically, morpho-syntactic descriptions of the words) and whose branches represent binary dependency relations between the words of a sentence. The finite verb is the root of a full sentence. For example, the structure of *A man was shouting dirty words* is shown in Fig. 2.

It can be strongly argued that dependency theory is an advantageous representation model for MT. Dependency trees of sentences are close to their logical forms and hence closer to their meaning than the corresponding constituent trees. We do not delve into the matter here in more detail (see Schubert, 1988 for a discussion on dependency theory and MT, and Melchuk, 1988, and Starosta, 1988 for general discussions on dependency theory). The dependency theory is applied in many other modern MT systems: DLT (Schubert, 1988) and Eurotra (Raw et al.) utilize it, and so do several Japanese MT systems.

Dependency structures have straightforward F-tree representations. If dependency relations are represented by their names in one feature in the dependent nodes, then an F-tree of a parsed sentence is a tree whose feature vector is a union of morphological, lexical, and relational features.

4 Translation

The MT Machine and dependency theory lend themselves naturally to a linear architecture of translation, When also each lexical transfer is described by a rule base, a possible system architecture has the simplicity of Fig. 3. That is in fact our implemented Finnish-English configuration. The MT Machine instances are marked with a special symbol.

The analysis phase includes morphological analysis (MA), dependency parsing (DP) and logical form reduction (LF). After DP and before LF data is converted into F-tree representation. Then the translation proceeds through several F-tree transformations: term and frozen phrase transfer (TT), domain-specific lexical transfer (DT), general lexical transfer (LT), structural transfer (ST), and feature transfer (FT). Then follows the synthesis phase which also utilizes the MT Machine: first the target tree expansion (TE) (inverse of logical form reduction) and then the target sentence production (SP). Each MT Machine application has its own

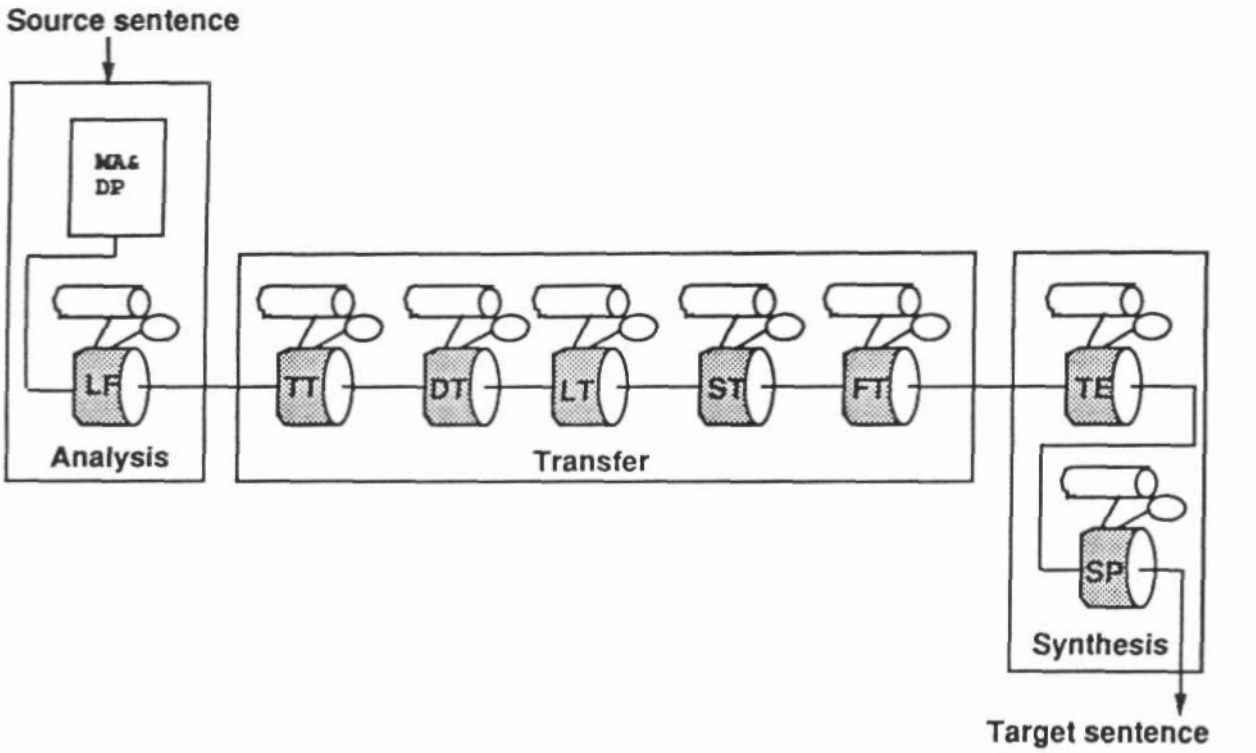

Figure 3: The translation process

rule base and each can choose its inference strategy independently from other phases. Notice how the sequence imposes hierarchy on the three lexical transfer phases.

The term "transfer" usually refers to projections between two languages that depend on both languages. Thus understood transfer is divided in our architecture into the subtasks shown in the figure. Transfer could of course be divided into subtasks in different ways. An administrative process implemented on top of UNIX and not shown in the figure controls the processes. It also includes tracing and debugging facilities.

5 MT Workstation

The translation architecture of the MT Workstations was shown in Fig. 3. The workstation also has to provide an interface for the external world. Interaction with the user takes place through a graphic interface. The screen is divided into input and output windows which display source language and target language sentences, respectively. Fig. 4 shows a copy of the two windows after the Finnish-English system has made a raw translation of an experimental text fragment.

The workstation concept takes post-editing seriously. One way of increasing translation quality in conjunction with positive user cooperation is to make editing and revising activities as convenient as possible.

The user can edit the texts in the windows in different flexible ways. He/she can move text fragments around or delete or insert new words using similar services as offered by modern text editors. If necessary, he/she can also tag sentences for later scrutiny.

Another important editing function is lexical replacement. It is a well known fact that one of the greatest problems in MT is the correct lexical choice. The rules of the MT Machine permit quite elaborate contextual checks in the lexical transfer phase. However, some pragmatic factors outside the text affect translation, and these facts are not within the reach of any rule system. The Finnish-English system features a dictionary of translation equivalents: Finnish words with sets of possible translations (in some contexts). If the user is not satisfied with a given lexical choice in the target text, he/she can point at the word and a window with a list of alternative translations will appear on the screen. If an alternative is pointed at, it will automatically replace the wrong word in the text - even in the right form.

The current Finnish-English system has approximately 5000 distinct lexical entries in the domain specific lexicon and some 35 000 entries in the general lexicon. Translation speed is a little over 1 word/second in VAXstation 3100 under ULTRIX. A speed which seems quite satisfactory considering that neither the MT Machine nor the rule bases have gone through optimizing efforts yet.

Only statistical tests based on real texts can give a reliable estimate of the linguistic quality of an MT system. We have carefully designed a testing procedure for the evaluation of our implementations. We are currently in the process of testing and thus for the moment lack sufficient data for precise judgements.

6 Conclusion

All human languages are open and complex communication systems, and it is generally agreed that no machine

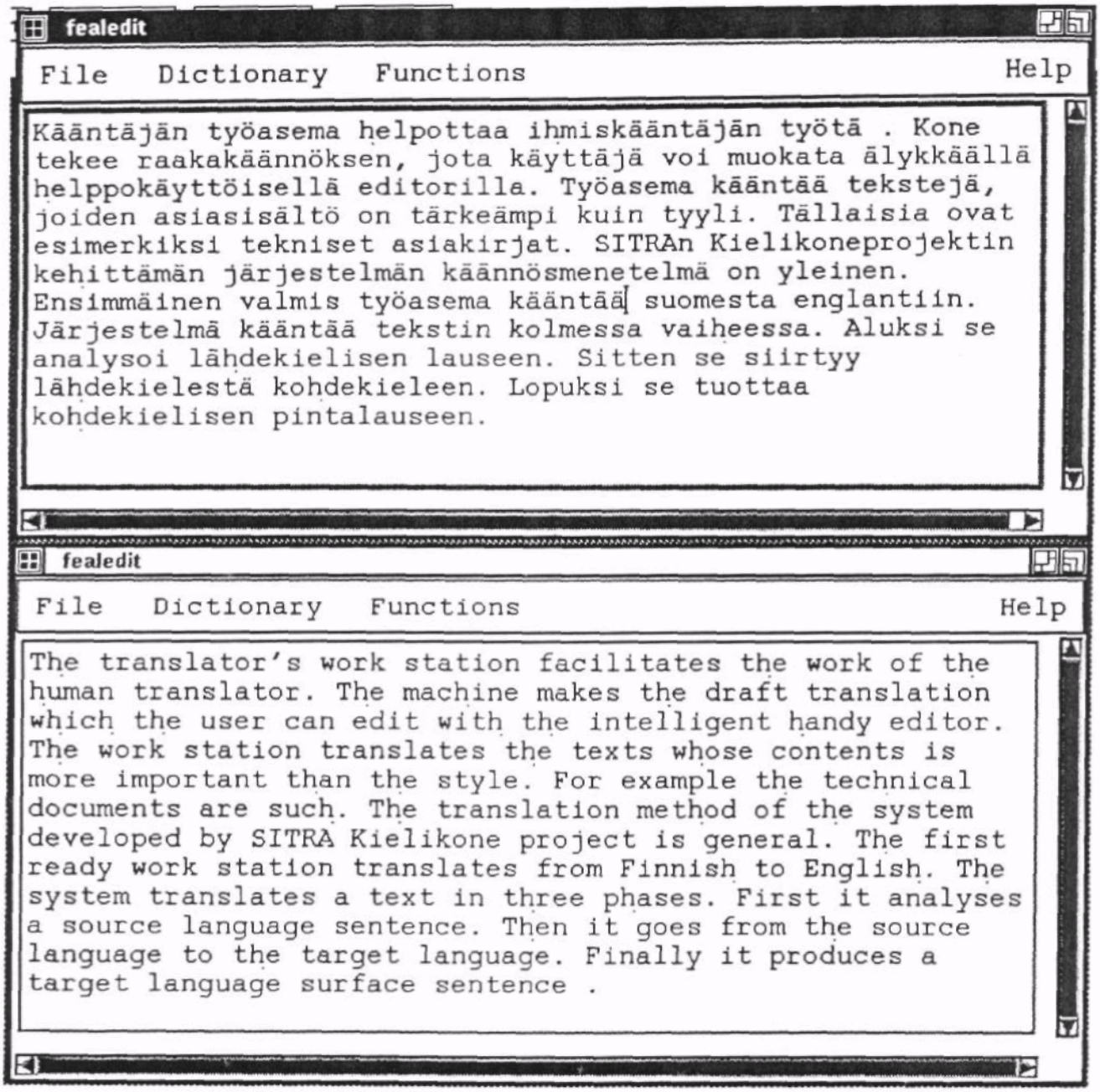

Figure 4: Source and target windows

translation system will ever be able to automatically translate all possible sentences from one language to another in high quality. One way to combat complexity and openness in language translation is to decompose translation into well-defined subtasks and solve each using declarative, modular rules. In the SITRA KIELIKONE project we have designed and implemented a general, language-independent MT Machine for the transformation of linguistic trees. Our full MT Workstation is composed of multiple, sequential executions of that machine. We have discussed a concrete implementation.

References

Jäppinen H. and Ylilammi, M., Associative Model of morphological analysis: an empirical inquiry. Computational Linguistics, Vol. 12, No. 4., 1986.

Jäppinen, H., Lehtola, A., and Valkonen, K., Functional structures for parsing dependency constraints. Proc. COLING86, Bonn, 1986.

Jäppinen, H. (Ed,), Synonyymisanakirja. Werner Soderström Osakeyhtiö, 1989.

Kulikov, L. and Jäppinen, H., Automatic translation of a highly constrained language. Proc. SCAI'89, Tampere, 1989.

Lassila, E., FORMO: program for Finnish word form synthesis. Proc. STeP-88, Helsinki, 1988.

Lassila, E., Parsing Finnish sentences by performing functionally defined sequential sub tasks. Proc. SCAI'89, Tampere, 1989.

Nelimarkka, E., Jäppinen, H., and Lehtola, A., Two-way finite automata and dependency theory: a parsing method for inflectional free-word-order languages. Proc. COLING84/ACL, Stanford, 1984.

Melchuk, I., Dependency syntax. In Melchuk, I., Studies in dependency syntax. Karoma.

Raw, A., van Eynde, F., ten Hacken, P., Hoekstra, H., and Vandecapalle, B., An introduction to the Eurotra machine translation system. In van Eynde, F. and ten Hacken, P. (Eds.) Working Papers in Natural Language Processing No, 1.

Schubert, K., The architecture of DLT - interlingua or double direct? Proc. New Directions in MT, Budapest, 1988.

Starosta, S., A Case for Lexicase, Pinter Publishers, 1988.

Takala, J., Pesonen, J., Kulikov, L., and Jäppinen, H., Machine translation for chemical safety information, International Labour Office, Geneva, 1990.

Valkonen, K., Jäppinen, H., and Lehtola, A., Black board-based dependency parsing. Proc, 10th IJCAI, Milan, 1987.