

Japanese-to-English Project

PROTRAN & TWINTRAN

Jelinek, J.¹, Wilcock, G.², Nishida, O.², Yoshimi, T.²,
Bos, M. J. W.³, Tamura, N.³, Murakami, H.³

1. Center for Japanese Studies, The University of Sheffield, S102tn, U.K.
2. SHARP Corporation, 492 Minosho-cho, Yamatokouriyama, Nara, Japan
3. Faculty of Engineering, Kobe University, Rokkodai, Nada, Kobe 657, Japan

1 Background

I.D.S. stands for Integrated Dictionary Systems. Its distinguishing feature is the integration of the bulk of grammar (= morphology, instructions for syntactic analysis, transfer and generation) into the dictionary.

Research on these lines, aimed at Japanese-to-English Machine Translation, started in the early 60's and found practical application as a tool for teaching monolingual English speakers to decode Japanese. Applications of this method to other language pairs have also taken place.

The IDS approach to Japanese-to-English MT found sponsorship from the British Government and ICL from 1984 as part of ALVEY (IKBS project no.25, carried out at the University of Sheffield, England in cooperation with ICL and Kobe University, Japan). When the Japanese to English part of the ALVEY project was successfully concluded in 1987, resulting in the creation of AIDTRANS, SHARP Corporation (Japan) concluded an agreement with the hitherto partners and took over further sponsorship of this research. This note is about the work carried out after that.

We have since achieved a working prototype of the sentence-for-sentence component known as *PROTRAN* and work now continues at Kobe University, under SHARP sponsorship, on the development of a textwide component (*TWINTRAN*) which could run on top of the existing model.

2 Sentence-for-Sentence Component: *PROTRAN*

2.1 Linguistic Rules and the Processing System

Our main task in the last year of research has been to reformulate the sentence-for-sentence Japanese-to-English system in such a way as to make the complete linguistic information explicit, which are executed by a processing system separate from these rules. The processing system is all programmed in Prolog and executes the linguistic rules by applying a function to each type of rule. This task has largely been achieved by now.

The linguistic information resides in the following sets of rules:

- 1) Japanese-to-English Automatic Dictionary (at present 32000 entries), held in a relational database with seven fields for each entry (combined key comprises the fields `Entry_word`, `Translation`, `Word_class`, `Entry_code` and `Continuation`; outside key are the fields `Priority` and `Semantic_category`).
- 2) Prioritised list of permitted juxtapositional links
Morpholexical analysis is executed by a linear chart parser utilising the fields `Entry_word`, `Entry_code`, `Continuation` and `Priority` from the dictionary database and the prioritised list of permitted links. This yields a set of morphological word class

strings, each of which maps the input sentence, evaluated for their juxtapositional suitability and their morpholexical suitability as the best of all obtainable dictionary mappings of the input sentence. This evaluation takes place in two tiers, first utilising the prioritised list of permitted links (to obtain morphological optimum) and then on the basis of the field “priority” of each entry (to obtain lexical optimum). Only the overall optimum mappings are passed on for further processing.

3) Morpholexical Grammar Rules (which are linear rewrite rules)

A set of linear grammar rules is applied to produce strings of syntactic word classes out of the original strings of morphological word classes.

4) Syntactic Analysis Rules (allowing parsing into trees)

A version of Bottom-Up Parser [4] is then used to execute the linguistic rules for Syntactic Analysis, resulting in a set of all allowable Japanese trees for the Japanese input sentence. We now have a prioritised version of these rules, designed to avoid carrying out the complete search in favour of proceeding with the best option only and coming back only if this option fails in further processing. Work is in progress at present, to be incorporated in *TWINTRAN*, to implement this stage in the form of demand-driven prioritised chart parser (whereby the chart parser is controlled by an A* algorithm).

5) Sentence Pattern Transfer Rules (specifying case-type word order transfer)

A set of functions applies the rules of Sentence Pattern Transfer (which are defined as Production Rules), moving verb-dependent case groups to their appropriate English word order positions, supplying default prepositions for each case group and creating default Subjects and/or Objects where necessary. Unlike all the previous stages, which are all largely divergent,

this process contains little divergence and prioritisation has not yet been introduced.

6) Substitution Rules (which deal with all remaining word order transfer, which at this stage is Entry-Specific)

Substitution is a set of functions executing rules which finalise the English word order down to the lowest level of trees, but the output remains in the form of trees. This process tends to output fewer alternatives than have been input, as some trees are liable to be eliminated.

7) Generation Rules (which produce English word forms)

Generation produces actual English sentences by scanning all tree node labels in post-order, activating Generation rules by node labels. It still preserves multiple user choices not only from amongst different sentence-level renderings of the input sentence but also from within sets of local alternatives within each version of the sentence.

2.2 Points of Convergence

It is common knowledge that the kind of exercise described above is bound to entail combinatorial explosion at several points from (1) to (4) if no measures are taken to prevent it. An inseparable part of our method is the reliance on the so-called Points of Convergence to overcome this problem.

A point of convergence is a point at which all alternatives so far listed have the same chance of success vis-a-vis what may follow. A selection of the best alternative(s), or a ranking of these alternatives as to their relative “goodness”, may therefore be carried out at each point of convergence, i.e. several times before the end of sentence is reached.

Points of convergence may be total (when all available alternatives at that point stand an equal chance of success) or partial (as between only some available alternatives). These points are found not only at the stage of linear grammar but also on the trees produced in syntactic analysis.

At each point of convergence the quantity of information passed on to the next process can be significantly reduced. The information left behind can either be dropped altogether (as is the case with less than optimum morphological representations) or be graded into ranks and wait in a queue on a demand-driven basis.

3 Textwide Grammar

A random sequence of sentences does not make a valid text, anymore than a random sequence of words makes a valid sentence (even though both phenomena may occur by accident). This is not primarily because such random sequences would not make a coherent sense; that would only put them in the same category as numerous properly formed and perfectly official texts, which just happen to talk nonsense. Natural language is able to express nonsense, on purpose or otherwise. Nonsense can be grammatical and can be translated. Random sequences fall down mainly because they tend to be formally incoherent, and formal coherence is another word for grammar.

There are formal rules determining text coherence and most of these rules have to do with the formal aspect of correferentiality. Certain structures are formally able to refer again to some items (assertions, events, facts, objects or persons) that have previously been mentioned in the same text. Unless this “referring again”, or correferentiality, happens quite often and sufficiently thoroughly, the text cannot be understood as one coherent linguistic entity and may end up looking like a random sequence of sentences.

The rules of grammar governing correferentiality are based on the theory of depredication [2] [3]. We have formulated these rules for the specific process of translating from Japanese to English. Their implementation also requires a fairly simple but robust semantic network, based entirely on only one type of semantic relation known as subsumption.

We believe that *TWINTRAN* will be able to demonstrate the functioning of textwide grammar in time for this conference.

A processing stage which would come up with only one definite “optimum” alternative at the very end is not yet implemented. Since the main prospective user is meant to be a monolingual Japanese, we envisage the need for interactive disambiguation based on reformulating the Japanese sentence in alternative Japanese renderings.

Acknowledgement

We wish to express our gratitude to Prof. Steven L. Tanimoto of University of Washington and Mr. Mikio Osaki, Mr. Shinobu Shiotani and Mr. Hitoshi Suzuki of SHARP Corporation.

References

- [1] Knowles, F. E.; Jelinek, J. and Wood, M. McG. : The ALVEY Japanese and English Machine Translation Project, Proceedings of Machine Translation Summit Conference, Tokyo 1987.
- [2] Jelinek, Jiri : A Linguistic Aspect of Transformation Rules, in Acta Universitatis Carolinae - Philologica, I (Slavica Pragensia, VII), Prague 1965.
- [3] Jelinek, Jiri : Construct Classes, Prague Studies in Mathematical Linguistics 2, 1966.
- [4] Matsumoto, Y. : BUP : A Bottom-Up Parser Embedded in Prolog, New Generation Computing, Vol. 1, No. 2, pp.145–158, 1983.