EXPERIMENTS WITH AN MT-DIRECTED LEXICAL KNOWLEDGE BANK

B.C. Papegaaij
V. Sadler
A.P.M. Witkam

BSO/Research
Bureau voor Systeemontwikkeling
P.O. Box 8348
3503 RH Utrecht
The Netherlands

Abstract

A crucial test for any MT system is its power to solve lexical ambiguities. The size of the lexicon, its structural principles and the availability of extra-linguistic knowledge are the most important aspects in this respect. This paper outlines the experimental development of the SWESIL system: a structured lexicon-based word expert system designed to play a pivotal role in the process of Distributed Language Translation (DLT) which is being developed in the Netherlands. It presents SWESIL's organizing principles, gives a short description of the present experimental set-up and shows how SWESIL is being tested at this moment.

Introduction

The DLT project [WITKAM 1983] utilizes an Intermediate Language (IL), in which form the translated text will be transported to the various receivers, where it will be translated into any of the available target-languages. DLT, therefore, is essentially a double translation - first into, then out of - the IL.

As a consequence of this strategy, one of DLT's central features is the pivotal role played by its Lexical Knowledge Bank. This knowledge bank, a central part of SWESIL (Semantic Word-Expert System for the InterLingua), has been designed to contain the lexical knowledge the DLT system needs: the entire vocabulary, contextual dependencies and all the system's semantic knowledge: the information necessary to distinguish between word meanings, knowledge about the way those meanings mutually influence each other, and the procedural knowledge that enables the system to calculate the probability of each possible interpretation of a given sentence (or fragment) and carry out the intricate process of lexical disambiguation and paradigmatic selection.

As both sides of the tranlation process greatly rely on the kind of knowledge SWESIL embodies, and since they both use the IL as medium for disambiguation, it appeared advantageous from the start to store SWESIL's knowledge entirely in IL format. This choice is supported by a number of properties of the IL such as its lack of homonyms, freedom from syntactic ambiguities, power of expression, and easy inspectability [see for details: WITKAM 1983].

The structure of the knowledge bank

The SWESIL knowledge bank is a multi-dimensional taxonomy, incorporating logical, ontological, and contextual dependency information. Its basic unit is the dictionary entry (Table 1). The starting point for each entry is an IL lexeme: a single IL word in its undeclined form. To discriminate between various intended usages of the one lexeme, and to link the lexeme into the taxonomic tree structure, it is given one keyterm for every meaning to be defined. This keyterm is a more abstract term for the concept the lexeme itself denotes, and relates to the lexeme through a kind of ISA link.

Table 1: An entry in the IL Lexical Knowledge Bank with added translations ({) Note the general structure: lexeme <keyterm> <<super keyterm>>, followed the depency pairs. Dependency pairs with more than one word on either side of the relator must be read as abbreviated forms for all the combinations of the left-hand and right-hand words.

konduk'i (to conduct)	<tra'ir'ig'i> (cause to go through)	<<tra'mov'iĝ'ig'i>> (cause to be moved through)

FIRST ARGUMENT	RELATOR	SECOND ARGUMENT
konduk'i <tra'ir'ig'i>	io-n (inanimate PATIENT)	kurent'o,likv'aĵ'o,gas'o (current,liquid,gass)
konduk'i <tra'ir'ig'i>	per (with INSTRUMENT)	drat'o,kanal'o,tub'o,il'o (wire,channel,tube,instrument)
konduk'i <tra'ir'ig'i>	ien-al,de (to/from PLACE)	lok'o,dom'o,maŝin'o,ej'o (position,house,machine,place)
konduk'i <tra'ir'ig'i>	por (to PURPOSE)	proviz'i,for'ig'i,el'ig'i (supply,remove,cause output)
konduk'i <tra'ir'ig'i>	ie-en,tra (within,trough)	kabl'o,sistem'o,kloak'o (cable,system,sewer)
inĝenier'o,instal'ist'o (engineer,installator)	as (AGENT of)	konduk'i <tra'ir'ig'i>
rekt'a (straight,directiy)	e (MANNER)	konduk'i <tra'ir'ig'i>
konduk'i <tra'ir'ig'i>	ie-en (into)	mar'o,lag'o,lok'o,ej'o (sea,lake,position,place)
for'ig'i,el'ig'i	io-n	likv'aĵ'o,gas'o
proviz'i	io-n	lok'o,dom'o,maŝin'o,ej'o

To ensure that the word is properly linked into the tree (and to reduce the number of possible search paths when disambiguating) the keyterm itself is labeled with a super-keyterm, related to the keyterm through the same ISA link that relates the keyterm to the lexeme. This information (the lexemes with their keyterms and super-keyterms) creates a tangled hierarchy structure, with the more specific lexemes towards the bottom of the taxonomy and the more 'primitive' ones at the top, with an unique upward path defined for each meaning. To provide inferential power, the IL intra-word grammar is used, by means of which SWESIL can decompose complex keyterms into their logical constituents and reason about them.

Each lexeme is described in detail in its entry: used both to differentiate the concept being defined from its "genus" (Cf. [CALZOLARI 1984, AMSLER 1980]), and to describe the contextual expectations of the lexeme in question. A definition is built up from dependency pairs: each pair consisting of two lexemes tied to each other with a relator. A relator is an IL word (usually a function morpheme or preposition) which denotes the roles the two lexemes play in relation to each other (see table 1). Most of the relators are used to represent the contextual expectation pattern of the lexeme: they specify the relations with the context typically expected for this entry, and the kind of lexemes most likely to partake in them. The complete information eventually to be contained in a definition can be said to represent what Mel'čuk calls the "Lexical Universe" of the entry [MEL'ČUK 1984].

The Disambiguation Cycle

The DLT syntactic parser generates dependency trees: structural descriptions of the syntactically possible representations of a given sentence [SCHUBERT 1986], from which a special diathesis module extracts relevant semantic information which it passes on to the SWESIL system in the form of dependency pairs similar to those found in the SWESIL entries. For each source language lexical unit, the diathesis module will search the SL to IL lexicon and will generate as many dependency pairs as necessary to capture all possible interpretations of a given part of the source string, using the process of paradigmatic extension (i.e. filling in, for each lexeme, all possible word senses found in the dictionary).

SWESIL receives those paradigms of dependency pairs (called IST pairs) and calculates which interpretation best fits its expectations by comparing them with the information in its knowledge bank (the SOLL pairs).

The actual process of selecting the best fit from a set of possible ones is one of ranking: the conflicting pairs will be ordered according to the height of their match score. Those pairs that best fit the knowledge bank information come out highest, those that fit less come out lower.

It is important to note here that the DLT system is designed to become an integral part of modern text-processing apparatus, and will parse texts 'on-line', starting the generation of 'parse trails' as soon as the first word arrives. The relatively slow speed of the typed input gives SWESIL the opportunity to 'interleave' with the parser and do a large amount of step by step pre-ordering, reducing the time needed for the final ordering (when all syntactically impossible interpretations have been weeded out) to a minimum.

The criterium for choosing one interpretation amongst various others is always relative: only when one interpretation scores substantially higher than any of the others can it be said to be preferred over those others and accepted as the one to be passed on to the final representation. Unless a definitive choice can be made, the conflicting pairs have to be handed on to the disambiguation dialogue, by means of which the human user of the system can assist in making the correct choice. Since SWESIL has already calculated the relative probabilities of the pairs, the dialogue module can use this to make the dialogue more intelligent and user-friendly, by presenting only the highest-scoring pair(s) to the user, not showing the full range of possibilities unless the proposed solution is rejected.

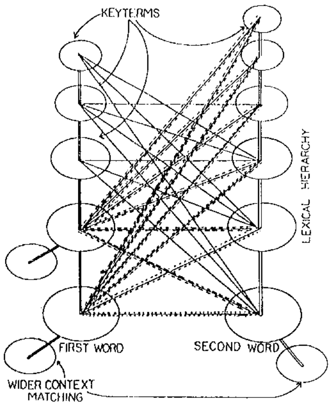
The Matching Procedure

To calculate the extent to which an IST pair conforms to the SOLL pair information in the knowledge bank, SWESIL uses a match score module. This match score module accepts as input one dependency pair, and returns a score which reflects how well that dependency pair fits the expectations found in the knowledge bank. What the match score module basically does is to take the input IST pair, locate the entries of both its constituent lexemes, and then search both entries for the occurrence of SOLL pairs that are similar (the notion 'similar' being defined by a number of matching rules) to the IST pair and calculate their measure of similarity. When both entries have been completely searched and certain boundary conditions have not been met (see below), both lexemes are replaced by their keyterms, and the resulting 'super' IST pair becomes the input to the same matching procedure (Fig. 1). Because of the way each lexeme can be recursively replaced with its own keyterm, the match score module in effect searches through the relevant part of the lexical taxonomy and records the match scores for the various levels of abstraction it reaches.

A number of boundary conditions prevent the system from falling into endless loops. The most important of these conditions is determined by the main purpose of SWESIL: to find among competing IST pairs the one best fitting the information in the knowledge bank. Because of this, SWESIL carries out the matching procedures for the IST pairs in parallel, and monitors the accumulated scores of each pair, testing on each level of matching to see whether one of the competing pairs has managed to score significantly higher than the rest. If this is so, SWESIL can 'freeze' the matching of the lower pairs and take the high-scoring pair as a 'working hypothesis', only 'unfreezing' the others if and when later evidence (after other sentence elements have been parsed) seems to invalidate the current one.

Fig. 1: By recursively moving up the dependency hierarchy, words can be matched at an increasing level of generalization.

Wider context matching can also (recursively) be applied to strengthen the first-order matches.



Another importa t boundary condition is inherent in the taxonomic structure of the knowledge bank. The recursive replacement of lexemes with their keyterms (which themselves consist of ordinary lexemes defined in the dictionary) inevitably moves the search-path upward through the hierarchy, leading to more and more 'primitive' lexemes. At some point the keyterms that replace the lexemes of the IST pair will themselves have no keyterms, simply because they are the most 'primitive' lexemes in the dictionary. In our IL dictionary, those primitive lexemes (the set of which we call the CORE) do not form a fixed set, but will undergo constant adjustment as the dictionary is developed.

In practice, CORE lexemes are those lexemes that have a markedly higher frequency of occurrence in keyterms or super-keyterms than other lexemes. This is in keeping with Wilks' observation about a frequency count of the Webster dictionary that the "fifty most frequent words show a remarkable overlap with the fifty-three defining elements [of his CSD system]" [WILKS 1972, p. 181], an observation which has been repeated by lexicographers like [AMSLER 1980] and [CALZOLARI 1984]. When the (substituted) IST pair lexemes can no longer be replaced by keyterms, the process must stop, since the match score table now holds all information available in this part of the taxonomy.

A third boundary condition, which has an important function in preventing endless loops, is the attenuation factor. This is a factor by which the score for a certain level gets reduced for each step this level is away from the 'entry level' at which the original IST pair entered the matching cycle. The further removed from the entry level (level 0), the lower the maximum possible score will be; and at some point the maximally obtainable score will fall below a certain threshold level, at which point SWESIL stops matching. In this way, the attenuation factor ensures that the system will eventually always escape endless loops, and it takes into account the distance between the literal IST pair as it entered the match score module and the level at which a certain match score was found (Fig. 1).

The Semantic Work Bench

The SWESIL system has now been under development for slightly less than a year, and it is still very much an experimental system. A large part of the effort is concentrated round the creation of the lexical knowledge bank. The present lexicon consists of ca. 1800 IL definitions based on 800 SL (English) words, which amounts to ca. 44.000 dependency pairs, and ca. 10.000 TL (French) equivalents. The number of English entries will grow to ca. 5000 within the next two years.

An experimental environment, the Semantic Work Bench (implemented in Quintus Prolog) is under development, in which the knowledge bank and the various decision mechanisms SWESIL uses can both be tested and developed further. In particular the SWB enables us to study:

a) the effect of different thresholds, match score calculations, order of searching etc., i.e.: given the information in the lexicon, how do the various matching parameters influence the process of disambiguation
b) the accuracy and power of dicrimination SWESIL can achieve
c) the adequacy of the dictionary entries and their usefulness in the process of lexical disambiguation, i.e.: can the SWESIL system really capture the knowledge and expertise of the lexicographer

At the time of writing, the first test runs have been completed, each involving the disambiguation of a single SL pair, extended to a number of alternative IL pairs (table 2).

A typical example of such a test run shows as input an English dependency pair taken from one of the test sentences. First, this dependency pair will be represented by several IL pairs to account for the different meanings of the SL pair, then SWESIL starts the matching procedure. The full output (not shown here) shows: the IST pair that is being matched, the lexeme (with keyterm) that is taken as starting point, the SOLL pairs that were found to match, together with their match scores and the hierarchic level at which they were found.

At the time these experiments were run, SWESIL had no other information to work with than the single SL dependency pair. Because of the lack of wider context, it was not always possible to find a decisive difference between competing pairs. This merely means that SWESIL rates such pairs without context as being equally possible. Only at a later stage, when wider context is taken into account, will SWESIL be able to make a more confident choice between such conflicting pairs.

Later this year, the first of a number of tests set up in cooperation with Alan K. Melby (USA) will be carried out. In these tests, SWESIL's translation of English text fragments will be compared with High Quality HUman Translation, with the emphasis on the precision of lexical transfer.

Table 2: From English sentence fragments, IST pairs are generated which reflect their possible interpretations. The system then calculates the appropriateness of each pair, which is reflected in the score table.
(N.B.: These examples are excerpts from longer and more detailed lists which will become available later this year.)

1. "development of the capital"				
1.1	kapital'o	is	evolu'i	
	(financial captital has evolved)			- scored 1.750
1.2	kapital'o	is	kresk'i	
	(financial captial has grown)			- scored 1.407
1.3	majuskl'o	is	evolu'i	
	(capital letter has evolved)			- scored 0.000
1.4	majuskl'o	is	kresk'i	
	(capital letter has grown)			- scored 0.160
2. "economic expansion"				
2.1	ekonomi'o	as	kresk'i	
	(the economy is growing)			- scored 1.905
2.2	ekonomik'o	as	kresk'i	
	(economics is growing)			- scored 0.160
2.3	dilat'i	-n	ekonomi'o	
	(physically expand the economy)			- scored 0.105
2.4	dilat'i	-n	ekonomik'o	
	(physically expand economics)			- scored 0.084

REFERENCES

Amsler,R.A.(1980):The Structure of The Merriam-Webster Pocket Dictionary, Austin; University of Texas

Calzolari,N.(1984):Detecting Patterns in a Lexical Database, In: Proceedings of Coling '84, California; Stanford University, Association for Computational Linguistics

Mel'čuk,I.A. / A.K. Zolkovskij [Zholkovsky] (1984): Tolkovo-Kombinatornyj Slovar' Sovremennogo Russkogo Jazyka, Wiener Slawistischer Almanach, Sonderband 14

Schubert,K.(1986):Syntactic Tree Structures in DLT Utrecht, The Netherlands; BSO/Research

Wilks,Y.A.(1972):Grammar, Meaning and the Machine Analysis of Language, London; Routledge & Kegan Paul

Witkam,A.P.M.(1983):Distributed Language Translation Feasibility Study of a Multilingual Facility for videotext information networks, Utrecht, The Netherlands; BSO