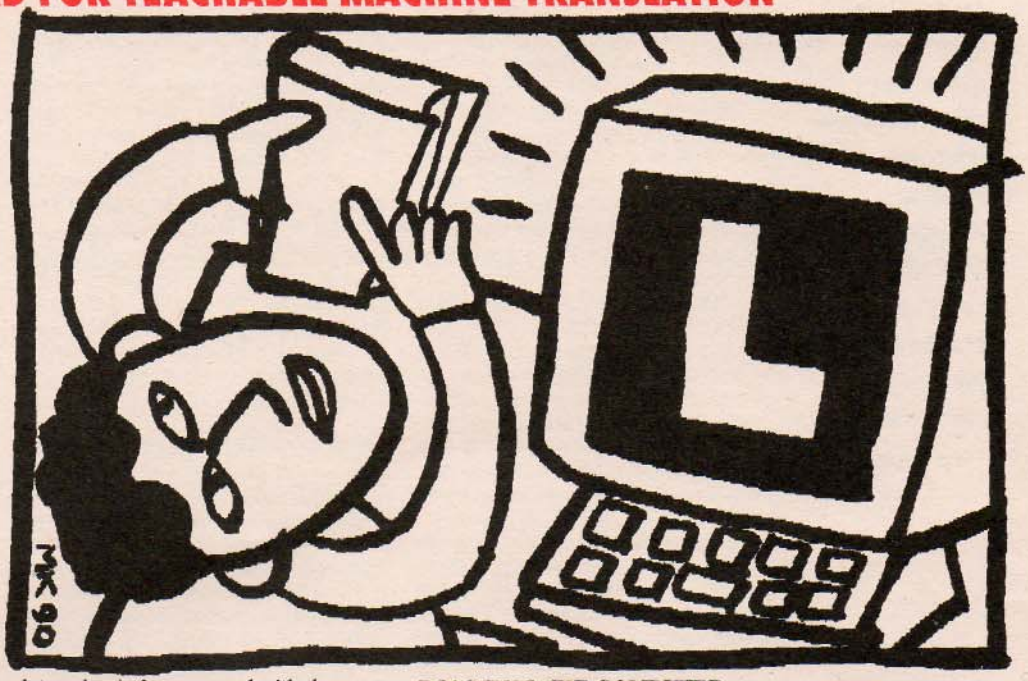
[Electric Word 18, 1990]

TOVNA – A NEW WORD FOR TEACHABLE MACHINE TRANSLATION



First presented to the world at the ASLIB Machine Translation conference in November 1987, the TOVNA MT system (whose name is a pun on the Hebrew words for "insight" and "software") has matured into a fully operational product, with a growing and prestigious user base in both Europe and North America.

Jerusalem-based TOVNA Translation Machines was founded in 1985. All development work is done in Israel by a staff of about 35. Chief designer is Daniel Cohen, active in the MT field since the late 1970s.

Currently, one language pair – English-French – is operational in both directions. Russian-English (also both directions) is in beta-test. And TOVNA claims that new languages can be developed on demand within 12 months thanks to the software's extensive modularity. Right now, Spanish, Italian, German, Arabic and Chinese are in development or under serious consideration for the future.

INNER WORKINGS

TOVNA is a so-called "transfer" system. The transfer takes place at a level quite close to the sentence's surface structure. There is no attempt to generate a deeper, more abstract representation (Eurotra or TAUM style) on which to effect transfer. Owing to the transfer approach, TOVNA uses a modular dictionary structure.

First, each language has its own lexicon (currently about 25,000 stem entries), listing all the words and describing their syntactic and semantic behavior with an elaborate set of categories. Each lexicon can be used both for source language analysis (parsing) and target language generation.

The lexicons are related to each other by two reversible bilingual dictionaries, one for single- and one for multi-word entries. Multi-word entries, or "phrases," are not applied blindly as simple word strings but described structurally.

Domain- and client-specific entries are included in the dictionaries and given priority during translation. The user can modify any entry – including verbs – except for several "highly grammatical" function words. And there are no size limits to either lexicons or dictionaries.

The most crucial feature of the dictionaries is a "bilingual phrase table" made up of transfer rules or patterns describing the differences between source and target languages (changes in word order, word additions and deletions).

You could, for instance, give the machine this example: *Sixty million US dollars/Soixante millions de dollars US.* Once fed into the machine, the pattern example can be tuned from the general to the specific. In other words, you can specify to apply the rule: only if the word is the same (*million* only); for any word acting as the same part of speech (*sixty* or any other determiner); for any word with the same semantics (*dollars or any word tagged "currency"*).

This is how the TOVNA system works. First, a *pre-processing* stage handles word hyphenation and formatting codes for typesetting. These codes will be re-inserted later into the translated text. *Morphology* and parsing are done in a conventional fashion. Several alternatives can be generated if necessary.

Parsing emphasizes analysis at the phrase rather than the sentence level, giving greater robustness to the translation. The alternative parses are based on hand-coded semantics (from the SL lexicon), system-learned semantics, and pattern statistics (see below).

Transfer into the target language is achieved by applying the transfer dictionaries and bilingual phrase table. Target-word ambiguity is resolved partly by domain preference and partly by using semantic codes.

For example, the English sentence *The machine works and the employee works* must be translated into French as follows: *La machine fonctionne et l'employé travaille.* In the first instance, works is translated as *fonctionne* because the subject is tagged as "inanimate;" in the second instance, the subject is "animate" and therefore triggers a different translation.

Finally, a *generation* phase inflects words, re-inserts formatting codes, etc.

Most of this is nothing new. Semantics and conditional phrase entries can be traced back to the early Seventies in commercial systems. Logos and ol' Systran currently make extensive use of such techniques. Whatever is new in TOVNA comes from its "learning from examples" strategy.

TEACHING THE COMPUTER

The "teaching" strategy reflects growing doubts about the ability of linguists to sit down and figure out everything the machine needs to know to process language properly – a daunting task indeed.

In fact, a recent trend among MT developers emphasizes the need not for greater linguistic depth but for alternative ways of describing languages without having to formulate everything. In other words, the tendency is to rely on "raw" linguistic facts rather than on their distillation by the human mind.

With TOVNA, "learning from examples" is done mostly through an interactive process of diagnosis, teaching, and retranslation.

First, TOVNA provides the user with a very interesting *diagnosis* function. As the system processes a sentence, it keeps a complete record of each successive step. When you find a mistake in the translated sentence, you can look up the record of its translation and discover the cause of the error.

For instance, you can examine the parse tree (shown by means of parentheses rather than graphically – rather a shame on a graphics-oriented machine like the SUN) and identify the source of the mistake.

Then you can go into the teaching phase by manually altering the parse tree. For example, if the phrase *the liquid oxygen tank* is wrongly translated as *le réservoir liquide d'oxygène*, the underlying structure will appear as "the (liquid (oxygen tank))." This does not make sense – a tank cannot be liquid – so you'll have to correct the parse thus: "the ((liquid oxygen) tank)."

Whatever you change in the parsing –

TOVNA SEEMS TO BE THE MOST PROMISING SYSTEM ON THE MARKET

THE PRICE QUOTED FOR EACH PRODUCT IS THAT CHARGED IN THE COUNTRY OF ORIGIN.

ON THE OTHER HAND, I AM CONCERNED THAT THE DEVELOPER MIGHT BE NEGLECTING BASIC FUNCTIONS (SUCH AS PARSING) BECAUSE OF THE LEARNING FEATURES – THUS PASSING ON TO THE USER THE ONUS OF SOLVING THE PROBLEMS.

whether it's a homograph or the structure – will be re-used through statistics (the "system history") and through semantics.

Statistically, the system will tend to remember that ((ADJ + N1) + N2) is more frequent than (ADJ + (N1 + N2)). And eventually, the system's default rules – those used when all else fails – will be altered accordingly.

Semantically, the system infers that liquid goes well with oxygen (and other semantically similar words) and will reapply this knowledge at the first opportunity without requiring you to code anything special. As the system gathers successful parses, it will refer to the semantics of the various words involved, and future parses having the same semantic relationships will normally be preferred.

You can also diagnose the transfer process by seeing which patterns from the bilingual phrase table have been applied. You can then teach the system by choosing another pattern or compiling a new one by means of a typical word sequence. The same logic applies to target word choice.

After teaching the system, you can launch an instant retranslation of the sentence, look at the result, and plunge again into diagnosis if necessary, until you're satisfied – or can achieve nothing more. This process, as you may imagine, is a labor-intensive one.

By teaching the system, you are post-editing your translation at the same time. But most users will want to perform extra polishing after teaching, for re-wordings which are too elusive or too long to teach.

You are not obliged to teach the system everything it translates. You may simply post-edit in the usual way, without any teaching. But in this case, you should exclude the text from the system's "history." The system's mistakes are not corrected internally and will continue to pollute the statistics. Such is the logic of a "learning" system.

Some of the learning is *automatic* – as is the case with parsing, already described. But learning through the bilingual phrase table is more of a manual process. You provide the examples and decide on their degree of specificity. So the figuring-out rests on you. Indeed, the process of teaching the system through examples requires a smart user who knows what he is doing.

USER CONTROL, PERFORMANCE

There could be some debate about the "learning"issue, but TOVNA definitely provides the user with a very interesting degree of *control* over the system's performance. This control will be welcomed by any user who has experienced the frustration of having to submit a special request at the bottom of a vendor's "priority list."

However, user control is largely focused on the transfer process. There is no direct control, for example, of the parsing process to enable you to change structural preferences or homograph defaults. The user can influence parsing only indirectly, through semantic learning and statistical data. Apparently, though, TOVNA is trying to open up this portion of the system too.

So far, TOVNA provides import/export filters for Wordstar and WordPerfect, as well as for phototypesetters. The developers claim that they can readily develop additional filters on demand. And an interface with the powerful desktop publishing system Interleaf is under development.

The various system functions are menu-driven and well arranged. For example, within the post-edit/teaching environment, the user has access to all dictionary update functions. They claim that coding a new lexicon entry takes 10 to 30 seconds, and an average-length multiword dictionary entry 30 to 60 seconds. Existing glossaries of the client's terminology can be imported into the system but nonetheless require extra coding.

Still, the TOVNA system is not simple to use. It is true that the "learning through examples" feature spares you the drudgery of writing out linguistic rules. But you still have to know what you're doing (what the general result to be achieved is, what the best way to specify it is, etc.). This kind of expertise or flair requires months to develop. User training takes between one and three days, depending on the level (normal or expert user).

Translation itself is a batch process; only teaching is interactive. On a SUN 3/60, TOVNA will translate between 1800 and 3600 words per hour, depending on such factors as sentence length and the number of concurrent users.

It is difficult to evaluate the performance quality of a "learning"system like TOVNA at short notice, because of the "buildup of knowledge" effect. In any case, the system looks to me somewhat immature – it being a well-established policy among MT vendors to start marketing the product before it is optimized.

I am also concerned that the developer might be neglecting basic functions (such as parsing) because of the learning features – thus passing on to the user the onus of solving the problems. All developers insist that the user must customize the system, which makes good sense. But sometimes, they tend to confuse this with basic improvements, which should be the developer's responsibility.

The World Bank in Washington is currently using TOVNA experimentally. So far, they report "mixed but encouraging results," adding that they are not yet assured of the system's cost-effectiveness. Their final conclusions will be the best indication of TOVNA's potential.

COST, HARDWARE

A perpetual licence for the software costs US$150,000 per language pair, based on a two-user SUN workstation. Maintenance fees run at 15% per year.

TOVNA is a power-hungry system but has avoided being dependent on large minis or mainframes. It typically runs on a SUN 3-60 workstation (about US$10,000-15,000). But since it is UNIX-based and programmed in C language, it could run on a high-end PS/2, Macintosh II, or NeXT (basic requirements: a 32-bit processor, 16 Mb of RAM and 300 Mb of disk space). Anyway, the cost of hardware in this performance bracket is bound to fall sharply in the coming years.

Multiple users can work at the SUN workstation by means of two microcomputers or terminals. A typical setup could be one SUN 3/60, one PC-XT, and one VT-200 terminal, with access to the system functions from both screens. Translations can be run in background mode while users perform postediting, teaching, or dictionary update tasks.

From this early viewpoint, TOVNA seems to be the most promising system on the market (let's exclude METAL, about which I can't really speak). It provides the user with a remarkable degree of control, and the teaching by example function certainly makes it userfriendly. In addition, it is equipped with certain self-learning capabilities.

Finally, using statistical history is a sensible means (when other resources fail) of selecting the most plausible option. Because of these innovative features, TOVNA deserves to be watched closely.

— Claude Bédard

TOVNA Translation Machines Ltd., 17 Beitar Street, Jerusalem, Israel. Tel: +972 (2) 719 157/623