Comprehension Assistance Meets Machine Translation

Gábor Prószéky
MorphoLogic
Késmárki u. 8, 1118 Budapest, Hungary
proszeky@morphologic.hu

Abstract

We present some basic principles around which comprehension assistance systems should be built, and then try to give an overview of how and in what direction the HLT industry should develop, in order to provide the best solutions to one of today's global market problems: to overcome the language barriers. Fully automatic machine translation (FAMT), as a discipline, is nearly as old as the computer itself. There was a time at the end of the 80's when many researchers felt that the breakthrough to real-life applications was very close. We have to admit, however, that that optimism evaporated rapidly. For sure, we have seen some interesting projects, and, in some limited areas, there are even functional implementations. In general, however, FAMT is still in its infancy, and, just as a human being, it may easily take a couple of decades more of gradual and concentrated development (rather than breakthroughs) to bring it up to maturity. Therefore we try to sketch something more modest than FAMT, but something, which is reality or is very close to reality today: human comprehension assistance. Recent tools of this category work mostly on the word level providing the user with some sort of dictionary information. We will show that a special sort of example-based machine translation can meet the comprehension assistance paradigm providing a new family of tools: sentence level comprehension assistants.

1. Introduction

A comprehension assistance program (Feldweg & Breidt, 1996; Segond & Breidt, 1996) does exactly what the term suggests: provides help to the human reader to understand or interpret the text displayed on the screen, when- and wherever the user needs it. Of course, comprehension assistance could also be associated with other media than text: voice, graphics or images, but of all these, text is by far the most frequently used for conveying corporate (or private, artistic) information, especially on the web, the most important source of information today. Thus this presentation is dedicated to text comprehension only.

Now, in the first years of the new millennium, human language technology (HLT) developers should aim at comprehension assistance rather than trying to generate a complete translation. And the best way to provide that help is by no means to emulate a print dictionary on the computer. Rather, as I am going to point out , a comprehension assistance system should use and process (all) the text, which appears on the computer screen. This digital text, which is fully and easily accessible, should be its primary input.

We will also present some basic principles around which such systems should be built, and then try to give an overview of how and in what direction the HLT industry should develop in order to provide the best solutions to one of today's global market problems: to overcome the language barriers. Each of the working principles to be presented is an abstraction or a generalization of the fundamentals of today's systems:

- a much more user friendly interface, which creates a qualitatively better working environment than the dictionary windows or the pop-up bubbles that have to be opened by a mouse click
- dynamic linking capability: the freedom of creating different interpretations of the text on the run
- processing the whole context of a word rather than taking a word in itself as the input of the system
- providing assistance only where and when it is requested rather than maintaining it all the time (the idea I call bridging the gaps of comprehension)

2. Working principles for a new generation of comprehension assistance systems

The use of today's applications is often cumbersome. The user always feels that the screen is never big enough. There are always too many windows on it. Those windows have to be opened, dragged and closed. Inevitably, the user is obliged to do a lot of operations, which have, essentially, nothing to do with the task he or she is working on. The comprehension assistance tool we want to design must be part of this already complex working environment.

Before going on, let us declare openly: normally, a linguistic tool is a software utility and not a core application. Our company, MorphoLogic develops HLT solutions, so for us they are central, but, for the business or home user, they are just utilities. This means that they cannot be the center of the universe. They should stay sort of in the background of the application, which the user is working with. The user wants to concentrate on that application and would not appreciate a tool, however helpful otherwise, that would force him or her to leave that program. Users want to have their environment intact, at the same time, they need immediate assistance, help on the fly. It would be awful for the user to have to interrupt reading the text, moreover, just at the point where comprehension assistance is needed.

As a consequence, the HLT tools of the near future should by no means be based on the print dictionary metaphor. There is no space on the screen to open two pages side by side as with a book (and, even if physically there were enough space to do so, the window of the core application must not be covered by another program, which was just meant to help). Also, dictionary lookup should be completely different in an electronic setting, providing a much better service than just displaying a full list of meanings for a single entry from a single dictionary, but that is an aspect that will be discussed in more detail later. But to break with the print dictionary metaphor is not that easy. We have used those dictionaries for centuries.

So far, in may ways, the history of HLT tools has been like the first decades of another great invention, the automobile. People were so much accustomed to horse driven carts that, for a long time, car designers could not avoid imitating them. It took more than a generation to arrive to aerodynamic, streamlined designs, which had no resemblance to the vehicles of our great grandfathers.

2.1. The interface: using the third dimension?

A well designed HLT tool must be very easy to use, which means it should require the least action possible from the user. It should not change the current screen layout, must provide assistance only at the point of the text and for the time the user requests it. A tool cannot divert its user's focus of attention from what he or she is doing – it is just a tool, which is supposed to help. The way we should do this is like turning on a torch to project the help text (the meaning, the interpretation) on the screen at the moment the user needs it and turn that *projection* off as soon as the user doesn't want to see it any more. This should be accomplished without demanding even a click from the user. It is like projecting an external information, which is present only in the third dimension onto the two dimensional world of the computer desktop.

How could we achieve this in practice? By displaying a pop-up bubble, which does not have to be invoked or to be closed by any specific user activity. Of course, this bubble also covers a small part of the screen, but as soon as the user releases it, it disappears automatically. Our company has already implemented this principle in a rather successful product line. In *MoBiMouse* (Prószéky & Kis, 2002), the translation bubble pops up just by positioning the mouse over a particular word on the screen. No click or keystroke is needed. The word in question can be any text on the screen: a menu item of an application or the title of an icon as well. The most recent version of the product is *MoBiMouse Plus* (based on combined advantages of two dictionary technologies: MoBiMouse and *MoBiDic* (Prószéky, 1998)). It also offers to assign a corner of the screen as a translation (interpretation) display area. It is like the mirror in the car: the user can always have a look what is happening in the background, how a particular part of the text is translated, but his or her attention can be focused on the main application. Popular pop-up dictionaries, like *Babylon*, *WordPoint*, *Clicktionary* and others (Prószéky & Kis, 2002) have been implemented based on nearly the same idea, but there is a difference: they have to be activated by mouse click(s) or special hot key combinations. When it comes to convenience of use, this difference is substantial.

You may also make the observation that Microsoft's recently introduced *smart tag* concept has a similar function, enabling the user to associate freely chosen information with the text. Tooltips in any program work along this line, too. In these two cases, though the no-intervention condition is satisfied, the big difference is that smart tags get built into the document, and tooltips are "wired" into the application, which displays them. Thus, in this case, the idea of the "external projection" is not implemented. Again, this is important, because tooltips always show the same text and that text is determined by the software manufacturer, not by the linguistic service provider or by any other content provider. The "link", the assignment between text and interpretation is static.

2.2. Phases of comprehension assistance

Comprehension assistance tool may provide different outputs, but they work along the same line. There are various clients which pass information to a server, which processes it and passes the results back to other clients for display on the user interface of choice – which, speaking about business use, could also be speech synthesis. The sequence is very similar to the human interpretation process. We also read the text, process it and then declare what it means. The comprehension assistance process consists of three main phases:

- Recognition: first, in a recognition phase, a client picks up the text to be analyzed, either optically from the screen or from the application or the operating system directly (we will return to these options later).
- Normalization: then, the linguistic server processes all this context, with the word pointed at in its focus, and provides and interpretation, which can by dynamically assigned.
- Search for content: this information is sent to the content server (dictionaries, lexicons, encyclopedias, etc.)
- Output construction: Finally, the result is presented to the user by a client – in a pop-up bubble, in speech, or through any other interface demanded by the user.

2.3. Dynamic linking of text interpretations

This has led us to the second design principle mentioned in the introduction. One of the key features of the coming generation of HLT tools should be the freedom of *association* (linking) between any part of any text and any kind of interpretation. In a different use or setting, the same part of the same text in the same application can have a different interpretation (translation).

This kind of dynamic linking is not like a tooltip in Word, for instance, which, as we have just pointed out, is static, wired into the application. It is not like a link on a web page (hypertext) either, because there, again, the associated text (the URL where the link "jumps") is fix, determined by the author of the page. Of course, a hyperlink is much more versatile than a tooltip, as it is built into a document rather than into a program. Without hypertext, the whole internet would not exist. Even so, I would say that hyperlinks, however useful, have become ripe for change or development. Given the size of the web, the user can be happy not to have a link on every word of a text. At the same time, the user may want to have a different kind of association assigned to a part of the text than the link chosen by the author. He or she may be interested in a different interpretation (translation or other) of the same word or expression. To give the user the freedom (or at least a degree of freedom) of choice in what kind of associations to assign to the text he or she is reading provides a new quality of information, or comprehension assistance service.

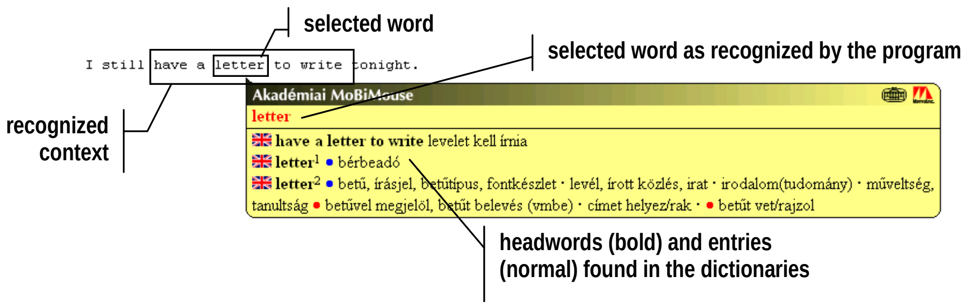

Figure 1: Parts of a dictionary entry as displayed by the comprehension assistant in the bubble.

External projection of information (our first design condition) and dynamic interpretation assignment are two closely related and equally important features we should expect of the new generation linguistic tool. Again, to use the example of MorphoLogic's MoBiMouse, the pop-up bubble of the dictionary program can show much more than just dictionary entries. A bilingual or explanatory dictionary is only one of the many databases that can be dynamically assigned to the text through MoBiMouse. The utility could also show stock exchange information assigned to company names, for example.

2.4. Assistance based on context

Probably the kind of association the user will want to make with a certain part of the text depends on what the *whole text* is about. Of course, one of the options is to select a target language to translate into. But the user may need different interpretations for business news and an engineering text.

By a definition, the meaning of a word is its rule of use. In turn, the rule of use is defined by the combinations and structures of other words, which the word in question can be used with. Thus a word in itself hardly has a unique, non-ambiguous meaning. Going to the extreme, we can even say that a dictionary is a very unfortunate abstraction. It isolates the words from their contexts and tries to give explanations to them in themselves. Sometimes it adds some auxiliary terms such as "slang", "colloquial", "informatics" or "biology", but in most cases, it provides a more or less arbitrarily sorted list of meanings, but there is no rule to tell when, in what contexts a certain word can occur and what its meaning could be.

If we see the word *"also"* by itself, there is no way to tell whether it was meant in German or English – or if it is a mistyped Hungarian word, which does not have the accent on the "*o*" (*alsó* means "under" in Hungarian). But if it is followed by the sequence *sprach Zarathustra*, there is no doubt that this *also* is in German, so I can assign a meaning to it in the above context.

It cannot be emphasized enough that words normally, in real life appear in a context. Even if I can see just the words *My computer*, it has a very well defining context if those two words are displayed under an icon on a Windows desktop.

2.5. Instant help for everyone

It is very unlikely that a web surfer would open a page, which is in a language that he or she does not know at all. The user normally has at least a basic knowledge of the language (English, to take the most common example). He or she can proceed in the text with no or little problem up to certain point. Then, after asking for assistance, will be able to go on, let's say, 27 sentences more. At that point, he or she will ask for assistance again. This is what I call *bridging the gaps of comprehension*. It is like jumping from one dark point to the other, turning on the light of the "projector" at those points only, providing instant assistance, and then letting the reader go on his own. Many times, very little help is enough. The human reader may know a word but not recognize the fact that, in a particular context, it forms part of an expression. But the machine may have that knowledge and help the reader out in that very moment, requiring no action from the user.

Even professional translators can have a similar demand, mainly if professional terminological dictionaries are also simultaneously available. They do their job, but there may always be a special constellation, which can hold them up for minute. A comprehension assistance tool, which is designed according to the principles of minimal user action requirement, projector style interface, and context analysis, with instant help display, will make the professional translator's work much more efficient.

3. Pattern-based machine translation

If we see what's going on in the field of machine translation, we can meet a lot of example-based machine translation (EBMT) systems. EBMT was suggested in the '80s as an alternative approach to rule-based machine translation. Since then, several authors have pointed out that the performance of EBMT can be considerably improved by adding linguistic background knowledge to the system (McTait, 2001). A reasonable goal is to find an optimum between EBMT and RBMT in terms of practical applicability: translation quality and speed. EBMT is mostly considered a statistics-based, probabilistic process, whereas RBMT is often thought of as a fixed, traditional, deterministic approach. In contrast, according to our view, EBMT and RBMT are just the two extremes of a generalized model. If we take "examples", they are not necessarily directly extracted from corpora, or produced by statistical analysis. Rather we opted to build a database of structural segments (cf. TAG or treebanks), which have been generated from various sources: lexicons, dictionaries and corpora. These automatically generated items should be complemented by manually produced structures. These structural segments can be referred to as *patterns*. These patterns show some similarity to the translation patterns used in HICATS/EJ (Kawasaki et al., 1992). Their translator knowledge base consisting of patterns is mainly utilized to translate idiomatic or nonstandard expressions. Our approach treats patterns as basic tools for describing both standard and idiomatic behavior of sentences, clauses, phrases and—what's more—lexical information. Shorter patterns are close to the *lexical items* of existing theories, fully specified multi-word items are called *idioms* in other systems. If a pattern comes with all of its attributes specified, then it is an *example* in traditional terminology. If no attributes of a pattern are specified, then the pattern becomes a *rule*. Thus both examples and rules are considered as special patterns. Our approach puts the emphasis on the transitions between the two: on patterns that have some elements filled in, but which are not fully specified. A key issue in the proposed model, called *MetaMorpho* (Prószéky & Tihanyi, 2002), is how to manage these generalized patterns.

First the input is segmented into words, which will serve as the actual input sequence (terminal symbols or tokens) of the parser. The morphological analyzer determines the attributes of these symbols, including their lexical form, case, conjugation etc. Second, the a syntactic parser analyzes this input sequence but it works with pattern pairs instead of rules: for every pattern for the source language, there is a set of corresponding patterns that determine the local structure of the target tree immediately below that node. This is called the interpretation of the source tree, which is, as opposed to the analysis, a top-down operation. For every root symbol created, a target tree is built following the structure of the source tree and only altering it within the scope of a single node at a time. The terminal symbols at the leaves of the target tree are fed into the morphological generator to produce the output sentence.

The MetaMorpho approach is, hence, neither direct nor interlingual, and it is also opposed to transfer solutions in that there is no need to transfer an abstract structure at any level in the original sense: analysis is done with the final output in mind, and can produce the result in a very straightforward manner (Figure 2.)

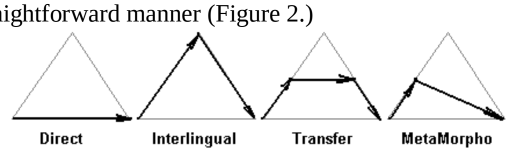

Figure 2: Strategies of MT solutions

4. Machine translation as sentence level comprehension assistance

Meaning is nearly always context-dependent. Each word in a sentence, which consists of a context of *(n-1)* words – all the rest of the sentence. In order to provide the correct translation of the sentence, the meaning of each of the *n* words has to be defined in the context given by its other *(n-1)* companions.

The above sketched sentence level comprehension assistant, MetaMorpho (now in beta testing phase), has been designed to work combining non-statistical EBMT and context-handling sketched in the previous sections. Whenever its set of patterns (rules) allows it to do so, MetaMorpho produces the translation of a complete phrase or sentence, taking into account the mutual context of each component word *(Figure 3).*

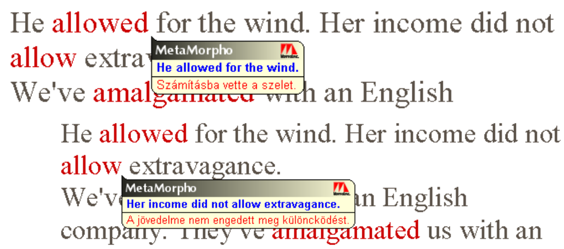

Figure 3: Context-dependent translations

If this is not possible, it works as a special dictionary program trying to list only those meanings of the word or expression in question, which are adequate to the context. That is, the MoBiMouse technology (see 2.1) is used.

MetaMorpho has been tested for English-Hungarian translation, and showed very promising results both in translation quality and speed. The number of the so-called core type patterns for English is surprisingly low: around 1,000. These basic patterns serve as examples for the more specific ones. Lexical patterns have been derived from existing lexicons and various collocation databases. This is a crucial part of the project since the building of such an inventory of lexical patterns from scratch would require several man-years. As opposed to the low number of core patterns, the number of lexical patterns is well in the hundreds of thousands. Assuming that one pattern can be stored in 100 bytes and a typical PC can be expected to have 100 megabytes of free RAM, the maximum number of patterns that can be used by the system is around 1 million.

We have just spoken about getting to the sentence boundary, implying the translation of whole sentences, and, in this way, a whole text. Normally, when people think about translation, this is exactly what they mean. If a human translator translates only fragments of the original, probably will not get too many more jobs. This is the ultimate goal of machine translation research and development, too, but to get there will take its time.

5. References

Clark, B., 2000. MoBiMouse, the first no-click translation tool. *Journal of Language and Documentation,* Issue 3, January 2000, pp. 26–27

Feldweg, H. and E. Breidt, 1996. COMPASS – An Intelligent Dictionary System for Reading Text in a Foreign Language. *Papers in Computational Lexicography (COMPLEX 96)*, Linguistics Institute, HAS, Budapest, pp. 53–62.

Kawasaki, Z, F. Yamano and N. Yamasaki, 1992. Translator Knowledge Base for Machine Translation Systems. *Machine Translation* 6(4), 265–278

McTait, K., 2001. Linguistic Knowledge and Complexity in an EBMT System Based on Translation Patterns. *Proceedings of the Workshop on Example-Based Machine Translation.* http://www.eamt.org/summit VIII/workshop-papers.html

Nerbonne, L. Karttunen, E. Paskaleva, G. Prószéky and T. Roosmaa, 1997. Reading More into Foreign Languages. *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP–97)*, Washington, pp. 135–138.

Prószéky, G., 1998. Intelligent Multi-Dictionary Environment. *Proceedings of the COLING-98*, Montreal, pp. 1067–1071.

Prószéky, G. and L. Tihanyi, 2002. MetaMorpho: A Pattern-Based Machine Translation Project. *Translating and the Computer 24,* ASLIB, London, in prep.

Prószéky, G. and B. Kis, 2002. Development of a Context-Sensitive Electronic Dictionary. *Proceedings of EURALEX*, Copenhagen, Vol. I., pp. 281–290.

Segond, F. and E. Breidt, 1996. IDAREX: description formelle des expression à mots multiples en français et en allemand. In: A. Clas, Ph. Thoiron and H. Béjoint (eds.) *Lexicomatique et dictionnairiques*, Montreal, Aupelf-Uref.

Turcato, D. & F. Popowich, 2001. What is Example-Based MT? *Proceedings of the Workshop on Example-Based Machine Translation.* http://www. eamt.org/summitVIII/workshop-papers.html