I Congreso Internacional de Traducción Especializada

MACHINE TRANSLATION AND TRANSLATION MEMORY: BREAKING THE BARRIERS

Yves Champollion
Traductor Público

Machine Translation and Translation Memory: Breaking the Barriers

Yves Champollion
Traductor Público
44 rue Danton 94270 Le Kremlin Bicetre, France phone/fax +33 1 46 72 62 33

Abstract:

The author wishes to explore the current limitations in both Machine Translation and Translation memory applications, and proceed to open up new avenues of research that could break through the current stalemate in computer-based translation.

Keywords:

machine translation, translation memory, complexity, language compression,

Introduction

MT applications are actually simple engines. They make use of various corpuses and basic algorithms that compute translation much the way chess software works. Beside the classical Analysis-Transfer-Synthesis model, more recent strategies have been developed, like the statistical approach, aimed at solving ambiguous propositions, but with limited success.

Some further progress is conceivable, by using better dictionaries, more complete sets of rules, better corpuses of irregular forms, and better analysis/synthesis software. However, the current MT model has reached maturity and will not get much farther on its own.

Translation Memory (TM), on the other side, is still in infancy. The basic models use databases of existing translations, which we call aligned documents, in both the source and target languages. Reduced fingerprints for each source sentence is stored in an index, and a fuzzy search engine scours the database each time a match is required.

TM is now beginning to exploit indexes based not on the bare textual contents, but on the structural, or syntactic, contents of segments. To exploit such a database of structural equivalencies, however, TM has to include reconstructivist capacities that are similar to machine-translation algorithms. When TM has not found a "content" match for a given source segment, but has found a "structural" match, the task will be to work on the triangular relationship (source segment to translate, database source segment, database target segment) to propose a target segment, using techniques akin to machine translation, based on dictionaries, rules, corpuses etc. Whereas machine translation works "blindly" from a source segment relying solely on its chess-like methodology, a "reconstructivist" approach, working in the triangular fashion described above, uses a human-produced guide of very similar structure (but with minor differences either in declension, gender, tense or terminology).

Obviously, both MT and TM have to join forces in order to bring the next generation of translation automates to life.

Limits of corpus-based translation

TM is known to perform well in vertical situations, where the new translation that is being undertaken is a sort of repetition of a somewhat similar and previous material. TM makes plenty of sense for corporations that have recurrent translation needs.

Suppose we deal with a project for which there is no previous TM. So, we gather whatever TMs we find and create a general-purpose TM, basically unrelated to the new translation job we have. I call this particular, random TM a "blind" TM.

Blind TMs are known to perform so badly, it's sometimes better not to use them. One temptation would be to make up for the lack of relevance (the lack of "leverage" as we say in the industry) with size - make up for lack of quality with quantity. Could blind databases of huge sizes be of any help? Does size matter? Is TM useful outside the niche of purely "vertical", in-house applications?

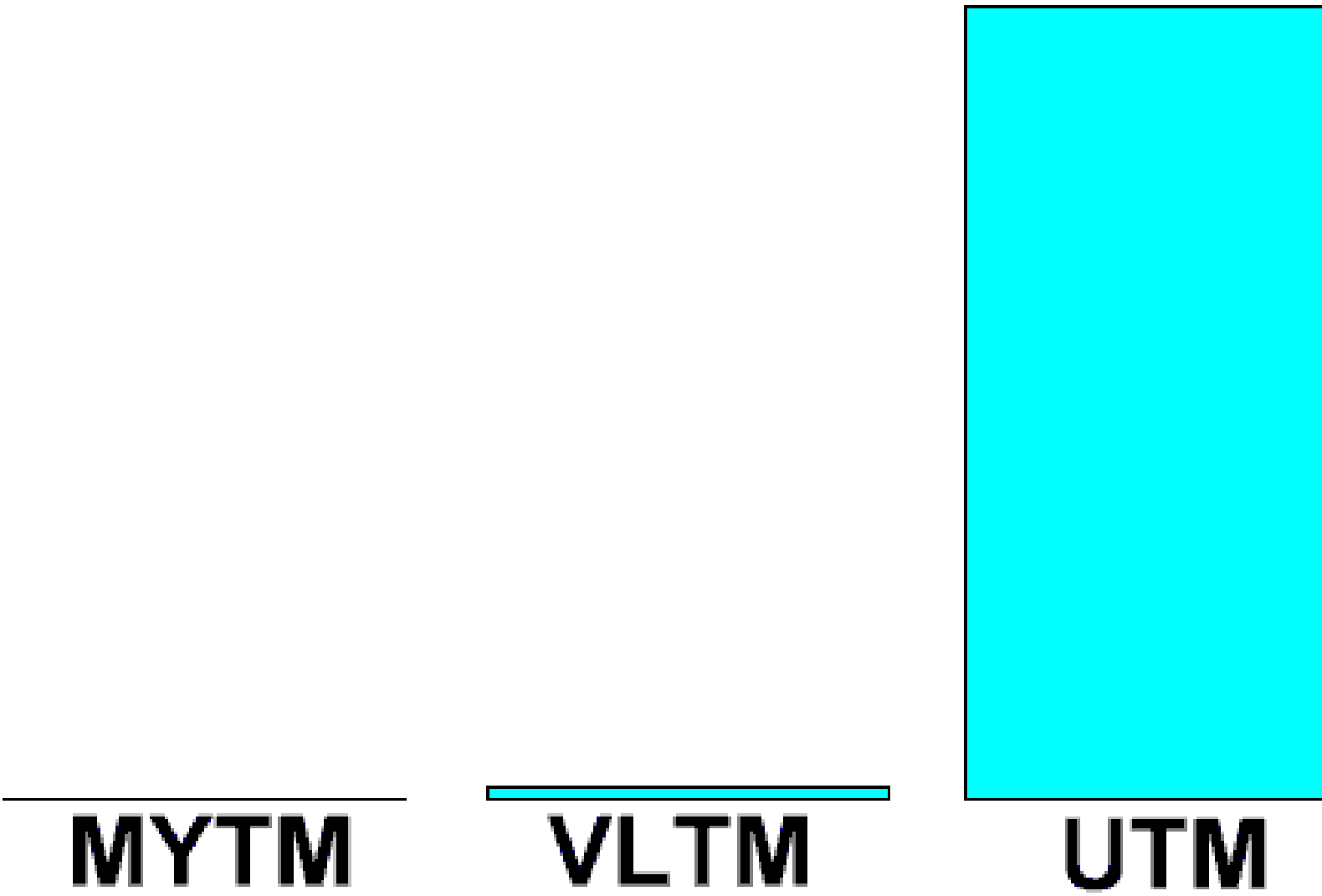
In the purely theoretical sense, yes. In the practical sense, no.

Suppose the perfect TM, the ultimate database. It contains all possible and sensical combinations of words a given source language can yield, each combination with a matching translation in the target language. This ideal TM would turn out a 100% match *every* time. This database does not exists yet and I doubt it will ever, but in our thought experiment, let's assume it exists and let's call it UTM for Universal Translation Memory. Each language pair has its theoretical UTM.

What sort of size would a UTM be? To effect a simple comparison, there are far more possible sentences combinations in the simplest of all languages than there are particles in the universe; and if a machine were to produce all combinations of words that make up intelligible sentences (if this were at all possible for a machine), using the fastest machines we have right now, the age of the universe would not be enough. A UTM for a particular language is utopia, but we will use the *concept* of it for our thought experiment.

Now let's take the largest TM that supposedly exists today on Earth. Let's suppose the Canadian government has neatly archived all English-French translations done in the past 200 years: 100 terabytes of aligned sentences. This looks awesome. On the other hand, compared to the UTM for that language pair, it's ridiculously small, perhaps one billionth of it. And I can predict disappointing, or nil, results if you use it to translate a recent and trendy Cosmopolitan article on Julia Roberts.

Let's take a detour to a branch of maths that’s interested in "Big Numbers" which also



deals with mind-boggling numbers and relies heavily on thought experiments. The number Pi (like other transcendental numbers) is supposed to have endless decimals, and they are supposed to follow each other in an unpredictable pattern - you'll never know the next decimal until you've calculated it. Now, is there a possibility for your last name (changed into ascii, or numeric, equivalent) to be clearly written somewhere in the decimals of Pi? The answer is yes, since it's an infinite sequence or random numbers - somewhere, your name is clearly written (and, just as many times, your name is also written with letters in the wrong order - unreadable "noise"). The problem is spending enough time finding it. But it's there.

To take things one step further, we could ask: is the Bible (Revised King James' version) written in clear form somewhere in this infinite sequence of decimals? Mathematicians are forced to answer "yes", otherwise, the word "infinite" loses meaning. Of course, the same text would be written innumerable times in the wrong order too. That makes a flabbergasting number of decimals in which to look for the right "signal", discarding huge numbers of "noise".

In short, all the knowledge of the Universe is already written - even in each one of us - the only problem being, how can we find it. Here we see the luminous intuition of Socrates ("Know yourself"), but this is not our subject.

The question narrows down from "Does the right information *exist?*" to "Can we *find* the right information?" For, if the Bible, or anything, is in the decimals of Pi, the hard question is, how do we find its location. **Finding it would take, even with the combined efforts of all men, mice and machines, far longer than the Age of the Universe.** Bad news. This is why, *theoretically*, size matters, but *practically*, it's a deadly mirage. You may have thought "why does this speaker take us to such philosophical quicksands to answer a practical question?" - but the answer is here.

To make things worse, the largest number of decimals of Pi computed to date with superlative computers is far from sufficient for my complete name ("y-v-e-s- -c-h-a-m-p-o-l-l-i-o-n") to appear in clear. King James may have to wait a lot longer.

Back to TM and reality. Our question was: can blind TM be efficient? As we have just seen above, the temptation of replacing relevance by bulk, or quality by quantity, is a dangerous one. You can keep adding aligned translations to a database, but you'll never get anywhere near a UTM, and the leverage increase will be very disappointing. "Big Numbers" Mathematics, however remote from reality it may be, teaches us at least this one point.

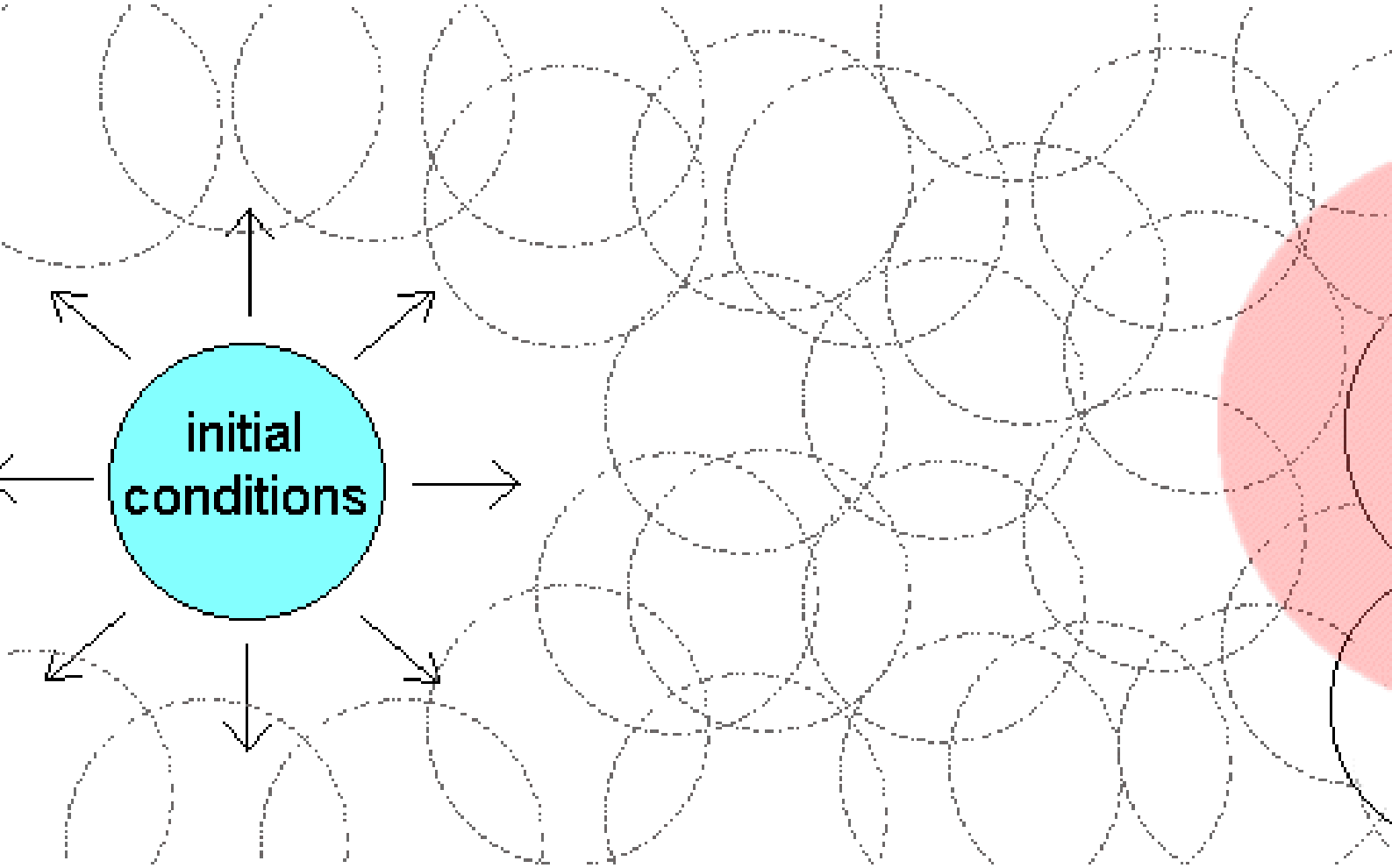
In the limited scope of a corporation wanting to build up leverage by archiving its translated material, growing a database does make sense. What I mean, however, is that these corporations should refrain from adding exogenous material, because it will give practically no added leverage - it will surely choke the existing database in the long run. I have seen a few people managing such corporate databases give in to the temptation of "more is better", frantically adding aligned material of dubious relevance (if not quality) to their databases and having to regret it later.

Complexity and Machine Translation

Each language has its own structure. And there is no way, from one language, to automatically guess the structure of another language, even if it is related. So we have an element of both complexity and chaos here and we can ask ourselves: can the recent insights developed by the science of Complexity (and its unruly sister, known as Chaos theory) help us here?

Note that this discussion focuses on the challenges of automated translation, not directly on classical linguistics. I am aware that related languages share similar structures, and that, at least within some languages, some rules allow the modelization of the evolution of some elements of language. But languages are primarily influenced by local conditions, evolving in unpredictable patterns.

The science of Complexity made one interesting discovery. Take a system with a set of initial conditions, let it evolve at random, keeping an eye on it. Observation shows that, with a particular set of initial conditions, even if the evolution follows random paths, definite patterns appear, and some sort of stability or equilibrium is quickly reached, at least for a while (like an eco-system). This plateau will remain for some time, then perhaps crumble rapidly, then again evolve into another plateau, another equilibrium. But the number of such equilibria, as compared to the nearly infinite number of theoretically possible combinations, is *small* and this is the key point.



The one striking point in the observation of chaotic systems in evolution is that, of perhaps a billion different sorts of possible structures they could evolve into, only but a few of them are evolved into. This becomes clear if you repeat the experiment from scratch many times, as this is now possible with computed "life games"; but it also becomes evident for all who studied the evolution of complex phenomena in the natural or human worlds.

Chaos theorists call this phenomenon attractors or even "strange attractors" because they defy explanation. In other words, out of zillions of possible "life forms" a chaotic system could evolve into, only a handful of them are actually evolved into, as if the chaotic evolution was "attracted" to some forms of organisation rather than others.

A language can be assimilated to a system in evolution. Even if all languages come from a proto-language, at some point, with different human groups going separate ways, and facing different "special conditions", languages evolved into what they are now (and will continue to evolve).

"Strange attractors" are also found in the realm of linguistics. Even facing different conditions, languages just cannot evolve into "any" form - they're bound to stick to a certain limited amount of possible structures. What is amazing is not how *different* languages are, but how *similar* they are. This is especially true when one thinks how far apart they could have ended, if “attractors” did not keep them together.

This is where it gets interesting. Even if every language has its own structure, the collection of existing structures is not a random collection, it is not chaos. Theoretically, it

is chaos, but in reality, only a very limited subset of all possible linguistic structures have been developed. Furthermore, there are reasons (which escape us at the moment) for which certain structures are not *possible* at all, and other reasons for which certain structures are *likely*. What research needs to do is to transcend the traditional categories of classical linguistics and move into meta-linguistics with the help of recent breakthroughs in both computational methods and the analysis of complex systems. Understanding the *raison d'être* of these linguistic attractors, understanding what they are and what causes them can both enhance our understanding of meta-linguistics, and open up new avenues in the field of automated translation.

Equip a computer with the right software. The sort of software that can spot structures, make deductions from sets of resembling and/or repetitive patterns. A wide array of such software exist: they recognise patterns or forms in graphics. They can identify a face in a picture, which otherwise is merely a sequence of apparently random binary digits, discriminating between a face and a background. They read maps and guide planes by taking decisions, avoiding obstacles, choosing an optimal route. They can read vast amounts of data to spot patterns. Other programs, known as fractal engines, can take a hard look at a vast collection of elements, then work out a series of equations, or rules, that can render the same collection of elements in the same precise order: they "structured" – in the mathematical sense - what is seemingly random data and can from then on "reconstruct" it. Even the apparently borderline activity of search for extra-terrestrial life has designed surprisingly accurate algorithms that can differentiate, in an apparently totally random set of data picked up from cosmos waves, "signal" from "noise", where "signal" would be anything produced by an intelligent life form. No E.T. has cropped up yet, but the algorithmic that differentiates signal from noise has benefited from it.

I know this is mind-stretching so let's make a pause. Compare two sets of data: a collection of one hundred photographs of faces of people from all races. To a blind machine, each picture is a set of random bits (pixels). But the engineer can hard-wire a program to recognise a nose, eyes, hair etc in this chaos, although, in each picture, they are *different* and even perhaps tilted (the software, as in OCR machines, could sense the picture is upside down and make up for it).

Suppose we take large corpuses from 100 languages. I do predict that software, of the class described above, will finally be able to spot structures. These structures will perhaps have *nothing* to do with our common, academical descriptions of languages, but they could be very interesting structures to use in machine translation.

So, this sort of software, correctly set-up and unleashed on vast tracts of bilingual material, will bring out patterns that are invisible to the human eye or brain. Now repeat the experiment, not just on a bilingual corpus, but on a multilingual corpus, ten or twenty languages wide, and you're bound in the end to have an emerging set of structures that, even if they're difficult for humans to grasp, will nevertheless represent some hidden, underlying patterns – and patterns there must necessarily be, as we saw in the discussion about chaos and complexity.

Structure

I have talked about structuring corpuses, and many may wonder what "structure" means here. I have to apologise in advance – I am not a classical scholar, and the categories I use are not canonical at all.

What we usually call "structures" in languages are the structures found by the human mind: grammar, syntax, as we learn them at school, and in the higher studies of

languages. I do not refer to them. I know that a lot of research, using public or private funds, is done to enhance machine translation with the idea that an MT system should implement as many of these rules as possible, in always smarter ways. I respect these endeavours. I wish to present an alternative.

I will take but one of the examples quoted in the previous section. The science of fractals, which is as old as mankind, has taken a sudden and spectacular turn with the advent of computers. A fractal engine finds patterns in a collection of apparently random information (for a computer, human speech appears "random" at first). Fractals are actually mathematical functions, or well-organised *transformational rules*, so they're very interesting for us MT engineers. It takes a lot of processing power to produce them. Fractal compression, for example, can compress an image by a factor much greater than classical compression. It translates a picture into mathematical functions then later reproduces the picture by executing the functions. The important point is, the computer looks at apparently random data (*random* from its own viewpoint, since computers do not have eyes), and derives some sort of *artificial order* out of it.

Extracting order out of language - writing down grammar - has been done by humans for ages. The idea is to give computers a chance: parse large amounts of texts, and derive some order out of it.

Classical compression compresses *redundancy* out of data. Fractal compression *structures* data. I would say that attempts at classifying corpuses of aligned translations by using grammatical/syntactic categories of human origin is tantamount to classical compression. It's good, but there's immensely better.

At this point, one correction is needed. I refer many times to grammar as being the structure of a particular language. But we're interested in *translation*. So we're interested not in the single grammar of one particular language, but in the "transformational" grammar (***T-grammar***) of a given language pair. This may sound utopian at first - you may ask: does such a thing exist? Can we find a set of rules that, to some extent, allows the transformation of sentences from one language to another?

One disclaimer here before loud protestations of utopianism are heard. We're not trying to make an engine that would translate *every time*, *every sentence*, and *exactly every time*. We're interested in improving the current level of machine translation, which is very disappointing. Always remember this. We're not playing God, just trying to raise productivity. With this in mind, my answer is yes: a T-grammar, one for each language pair, is feasible, opening the door to the reconstructivist approach I mentioned earlier. Even if perhaps only 25% of the sentences in a document are candidates to reconstruction (the rest being classically MTed), well, a 25% increase of MT productivity is a huge improvement.

Take a vast amount of aligned translations. Use a fractal engine that has been optimised for linguistic material. Make it work over this mass of data. The engine will inevitably produce a fractal compression, i.e. find patterns (invisible to the human eye) that can be represented by mathematical functions. These are the tranformational structures, the T-grammar, I talk about. If you look at what a fractal engine produces (a mumbo-jumbo of arcane mathematical functions, in apparent chaos), you will probably understand nothing at it. The only point here is: does it work? It does.

Where human grammar finds perhaps 1,000 rules in a particular language, a fractal engine would find a million – a thousand times more. Human grammar is a lot more elegant. Fractal-produced grammar is not “economical”, not “elegant”. So what.

Where it gets fascinating is having computers themselves work on producing the T-grammar that in turn will equip MT software. This is the crux of the question. In other words, current MT systems are computer pumps that are primed with human material. The grammar, or even T-grammar, on which they rely is man-made (it's the adaptation, to programming logic, of categories, structures, rules, created by humans). It's like having a multimedia device that's part analogic, part digital.

True, the “transformational” grammar of a human professional translator could be immensely more concise, well-thought, well-organised, than the obscure “T-grammar” computed by a blind machine churning at terabytes of aligned material. But the immense advantage the computer has is the total absence of headaches, considerable speed, and no fear of processing immense volumes of data.

One of the doctrinal dead-ends which we have to break is the reliance on “human” grammar, (most machine translations are smart adaptations to computers of how human think languages are organised). We should not be afraid to rely also on computer definition of how human language (more precisely, transformation of a language into another) is structured. From this on, move into redefining machine translation altogether.

Of course, numerous hurdles have to be overcome. For example, one particular source segment will not necessarily always be translated the same way, depending on the context. And there are many other problems. To all these, there’s an answer, at the very least a statistical one, which is beyond the scope of this paper.

This section tried giving some clues as to how we could improve on the algorithmic side of the MT question.

The time when 50% of the translation load can be decently pre-processed by a machine (as opposed to less than 20% today, in the best of cases) is not far.

Compression

I will keep building on the "fractal" aspect - there are other aspects. I just addressed the question of how to enhance the "MT" approach of translation, the machine-translation algorithmic, using computer-made T-grammar. I will now try to deal with how this fractal approach can also help the "Translation memory" aspect of the question just as well.

Intelligent compression (generating indexes) is one of the keys to harnessing the power of very large arrays of aligned corpuses.

Suppose a fractal engine has succeeded in compressing a vast amount of aligned material. It has created it own T-grammar, a maze of transformation rules, impenetrable to the human mind, awfully complex and requiring vast processing powers to implement.

A *majority* of the raw material’s sentence pairs are described by a *minority* of transformational functions, the remaining minority being uncompressible: rare structures of speech, unfinished or broken sentences, obscure techno-jargon, etc, which we could call “noise” for practical purposes. We have a so-called ***power law distribution***: a fairly small

collection of rules actually describes a majority of the way sentences are transformed from a language into another. The computer may have produced one million rules after having crunched a terabyte of aligned material, but perhaps 100,000 of them apply to 90% of sentences pairs, all the rest being endless “special” rules made necessary by “noisy” sentence pairs that do not fit in any commonly found pattern.

Here I introduce yet another concept, widely used in the theory of complexity: power-law distribution.

If our aim is to increase the productivity of machine translation (and again, not playing God, i.e. not trying to create the perfect translation system) then we can discard the rules that were made necessary by “noise” and focus on the majority, on areas of more certainty.

We keep the minority of rules that were re-used many times to describe numerous sentence pairs. If we later fall on other data mines of aligned corpuses, we will find, after processing them, that (after eliminating the minority of “noisy” sentences), the new mass of aligned data will practically not increase the size of our “structural” database. This gets fascinating. You can keep piling up aligned data - our index does not grow in proportion of the database mass: passed a certain threshold, this index size remains practically flat. *We side-step the "quantity over quality" dead-end I presented earlier in this paper - but this is not the key point.*

Note that we never discard the “content” database (mass storage does not cost much these days). Our structural database, however, now works as an index pointing toward it. In front of every entry in the structural database, we keep an index pointing to the various implementations (original forms) of that structure, in the “content” database. But the key point is, when searching for matches, we first search for a *structural* match, i.e. on a database that is *very limited* in size; and with an intelligent indexing system, we immediately find, then retrieve, a top-ten (top-thousand, whatever: computers are not afraid of quantities) list of “content” entries that have the most similarities with our original sentence.

So we have on the one hand, the segment we want to translate (A), then we have a list of perhaps 1,000 "candidate" matching sentences (B1, B2 ... Bn). Three questions arise here:

1. Do we have an exact match? Yes ---> use it, end of search.

2. If not, then how do we find, among the many sentences that share a similar structure, the *one* that approaches our sentence the most?

3. Or - could *all* these sentences help us in producing the desired translation?

Question 2: my answer relies on a set of primitives that is a lot richer than traditional grammars. For example, human grammars have very few primitives - or essential categories, like "noun", "adjective", "verb" etc. Computers, on the other hand, are not impressed by complexity. So categories can be a lot more precise. The noun "cat" is a primitive in most languages (a masculine noun in French, a "weak", feminine noun in German, etc). In computerese, "cat" could be a multi-level category like

Being -> Living being -> Vertebrate --> Mammal -> Feline -> Cat

So the set of primitives that "prime the pump" of an computer-driven T-grammar can be made of 30,000 different categories, in a neat hierarchy similar to the numbering system for posts in the modern accounting system. So if there is *no exact match* for a given source segment, but there is a collection of *structural matches* using various other terms, the match which terms are the closest, hierarchically speaking, to the terms of our source segment, will be chosen. It's a lot better to re-use a fuzzy source segment that refers to a dog (the reconstructivist algorithm working on replacing dog by cat) than using a similarly structured segment that deals with a coffee-maker. (For perhaps the *target* language uses a different verb for animate, and inanimate, beings).

Synthesis

As explained in the introduction, current models (TM and MT) must unite. I do believe that machine translation must keep its central position as the rule-based computational foundation. But recent trends indicate that there is an increasing tendency to complement rule-based (or algorithmic) translation with increasingly vast corpuses: corpuses of expressions, of idiomatic constructions, specialised glossaries that are automatically loaded when a certain context is detected; and most of all, translation memories. My aim was to open a perspective on how to squeeze out as much as possible from translation memories, how to devise synthetic, or artificial, T-grammars, and show strategies for using databases of immense sizes by fractally compressing them.