L1: A quick introduction to machine translation for translators

Mikel L. Forcada[1,2]

[1]Departament de Llenguatges i Sistemes Informàtics,
Universitat d'Alacant,
E-03071 Alacant, Spain

[2]Prompsit Language Engineering, S.L.,
Edifici Quorum III, Av. Universitat s/n, E-03202 Elx, Spain

Crash Course on Machine Translation
IBL, Bulgarian Acad. of Sci.,
Sofia, 2–4 July 2014

Outline

What is machine translation? (2/3)

In a diagram:

SL text → Machine translation system → **TL** (raw) **text**

(no human intervention)

Outline

Crude models in MT may save work (2/5)

Compositional ("building") view of sentence translation:

- **build** a representation of SL sentence meaning
  - start with the meanings of words as building blocks
  - using syntactical groupings to determine how meanings are combined into more complex meanings
- then **build** a TL sentence from this representation (reverse process in the TL)

This is why translators can translate sentences they have never seen: they **build** their translations!

Crude models in MT may save work (4/5)

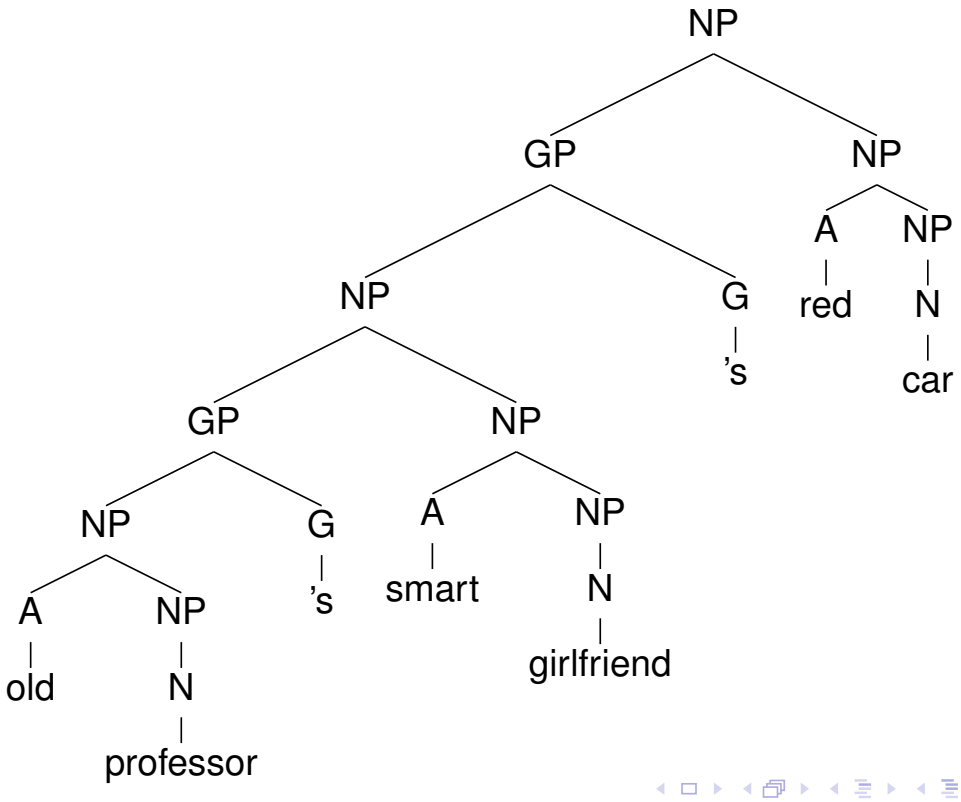
Approximation #2:

- Translate words to words (and multi-word units to multi-word units, taking care of terminology), . . .
- . . . transform SL word order and structure to TL word order and structure. . .
- . . . *et voilà*!

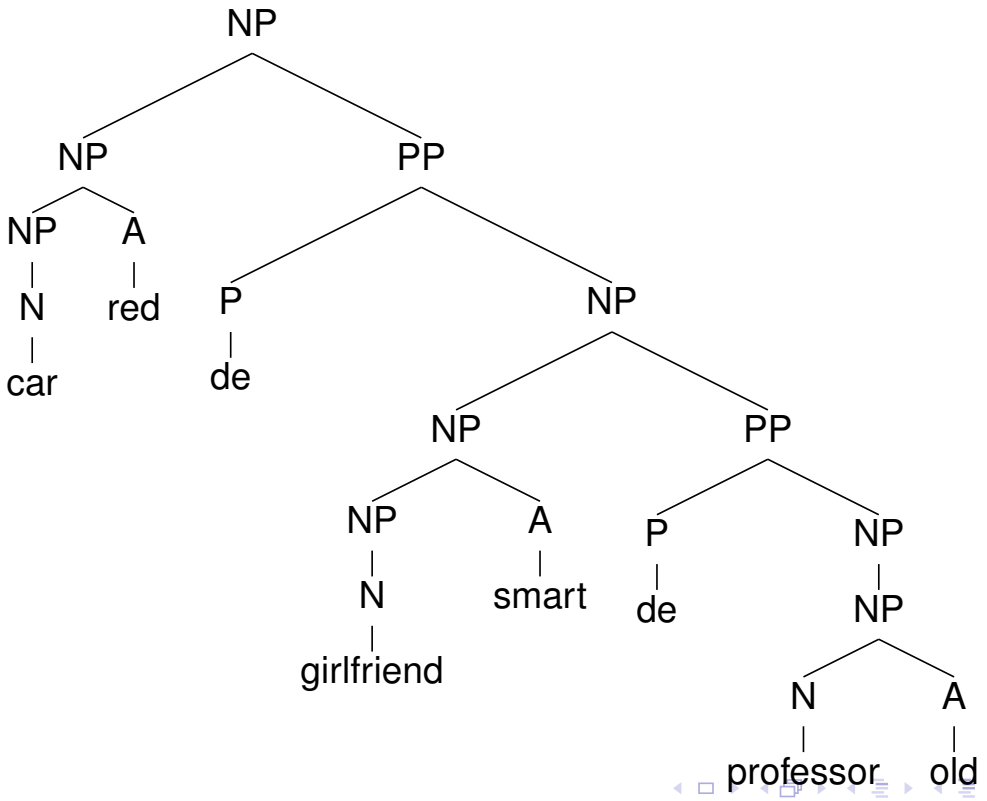
Usually called the "transfer" approximation.
No need to be a theoretical physicist!

Outline

RBMT: Source-language parse-tree



RBMT: Target-language parse tree



Statistical machine translation (SMT)

In statistical machine translation (Google, Bing, Moses. . .):

- Very large, representative sentence-aligned bilingual corpora (i.e., a very large set of sentence-sized translation units). . .
- . . . are used to train complex statistical models. . .
- . . . that (approximately) assign a probability to each possible translation of a new SL sentence.
- The most likely one (according to the model) will be the translation.

Professional knowledge is hard to code in these crude models (1/3)

Rule-based machine translation:

- builds upwards from word-to-word translation,
- hopefully to reach the sentence level,
- has trouble solving **ambiguity** at all levels:
  - lexical ("replace" → "put back"/"substitute"),
  - syntactical/structural ("I saw the girl with the telescope")
- Translators' intuitive, un-formalized knowledge about the task has to be turned into rules and encoded in a computable manner:
  - Additional crude simplifications and sacrifices needed!
  - If well chosen, some of them will often work fine.

Professional knowledge is hard to code in these crude models (3/3)

Statistical machine translation:

- Approximate probabilistic model trained on
  - sentence-aligned bilingual corpora
  - target-language monolingual corpora
- The use of target-language probability models sometimes leads to deceptively fluent, but inadequate translations (e.g., a negative word is lost!).
- Translations resemble those found in training corpora (opportunity for customization!)

Professional intervention is always needed (1/2)

- If “raw translations” are close to being fit for purpose (as they sometimes are), post-editing may be feasible
  - **Post-editor**: a **professional translator**, additionally trained to make the most of MT system output!
- If they aren’t close: post-editing is unfeasible, and professionals choose to translate from scratch
- How close? 80%? 90%? It depends on the language pair and on the posteditors themselves.

Outline

Machine translation: an ingredient in the mix (2/2)

Target-text-mediated interactive machine translation:

- As the professional types the translation. . .
- . . . MT system proposes completions which are compatible with what they have typed, . . .
- . . . and the translator either selects one or goes on typing [animation].
- **Examples:** TransType, TransType2; a similar one is CAITRA.
- All of them use *statistical MT*, but could use *rule-based MT*

Machine translation: an ingredient in the mix. . .

Target-text-mediated interactive MT (en→fr):

Source: This bill is very similar to its companion bill which we dealt with yesterday.
Typing the target: Ce

Machine translation: an ingredient in the mix...

Target-text-mediated interactive MT (en→fr):

Source: This bill is very similar to its companion bill which we dealt with yesterday.
Typing the target: Ce projet de loi **e**st

Machine translation: an ingredient in the mix...

Target-text-mediated interactive MT (en→fr):

Source: This bill is very similar to its companion bill which we dealt with yesterday.
Typing the target: Ce projet de loi **e**st **t**rès semblable **a**

Machine translation: an ingredient in the mix. . .

Target-text-mediated interactive MT (en→fr):

Source: This bill is very similar to its companion bill which we dealt with yesterday.
Typing the target: Ce projet de loi **e**st **t**rès semblable **au p**rojet de loi

Machine translation: an ingredient in the mix. . .

Target-text-mediated interactive MT (en→fr):

Source: This bill is very similar to its companion bill which we dealt with yesterday.
Typing the target: Ce projet de loi **e**st **t**rès semblable **au p**rojet de loi **qu**e . . .

Give MT a try!

Your mileage may vary, depending on your languages.

- Around 30% of freelancers[1] already use MT
- Don't assume general-purpose online MT is the best you can get.
  - It can only improve!
- Be patient: development can't keep up with demand!
  - It may be customized
  - You may help develop it
- Learn to post-edit
  - Machine translate your text, sentence-align it, and stick it in your favourite CAT program.

[1] See `http://goo.gl/nQBuV`

EU-rope needs machine translation!

Multilingualism is at the *soul* of the EU. MT can help!

- **24 of cial languages (so far!):** bg cs da de el en es et fi fr ga hr hu it lt lv mt nl pl pt ro sk sl sv
- **5 semi-of cial languages:** ca, cy, gl, gd, eu.
- **Many non-of cial languages:** an, br, fo, fry, lb, oc, . . .
- **Main immigrant languages:** ar, ber, hi, ru, ur, tr, zh.

(size $\simeq$ World total of speakers)

License

L2: Apertium: a free/open-source, rule based machine translation platform

Mikel L. Forcada[1,2]

[1]Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071 Alacant (Spain)

[2]Prompsit Language Engineering, S.L., Edifici Quorum III, Av. Universitat s/n, E-03202 Elx, Spain

Crash Course on Machine Translation
IBL, Bulgarian Acad. of Sci.,
Sofia, 2–4 July 2014

Free/open-source software

Software is **free** (Free Software Foundation, `www.fsf.org`) when

0. anyone can use it for any purpose
1. anyone can examine it to see how it works and modify it for any new purpose
2. anyone can freely distribute it
3. anyone may release an improved version so that everyone benefits

Conditions 1 and 3 require access to the source code, hence the name **open-source** (Open Source Initiative, `www.opensource.org`).

Commons

- **Commons:** a piece of land subject to common use: as (a) undivided land used especially for pasture or (b) a public open area in a municipality
- Analogously: **software commons**, code subject to common use

Free/open-source software: open for business/1

Free/open-source software opens new business models:

- emphasizes *service-centered* models over traditional *license-centered* models
- customers avoid *vendor lock-in* and may move into *technological partnership* with the provider of their choice

Machine translation software

- Machine translation is special: it strongly depends on data
  - *rule-based* MT (RBMT): dictionaries, rules
  - *corpus-based* MT (CBMT): sentence-aligned parallel text, monolingual corpora
- Three components in every MT system:
  - *Engine* (also *decoder*, *recombinator*...)
  - *Data* (linguistic data, corpora)
  - *Tools* to maintain these data and convert them to the format used by the engine
- For MT to be free/open-source, the **engine**, the **data** and the **tools** must all be free/open-source (NB, in corpus-based MT this includes **corpora**!)

Rationale /1

To generate translations which are

- reasonably intelligible and
- easy to correct

between related languages such as Spanish (`es`) and Catalan (`ca`) or Portuguese (`pt`), etc., or Nynorsk (`nn`), Bokmål (`no`) and Icelandic (`is`), or Irish (`ga`) and Scottish Gaelic (`gd`) one can just augment *word for word* translation with

- robust lexical processing (including multi-word units)
- lexical categorial disambiguation (part-of-speech tagging)
- local structural processing based on simple and well-formulated rules for frequent structural transformations (reordering, agreement)

Rationale /3

- It should be possible to generate the whole system from linguistic data (monolingual and bilingual dictionaries, grammar rules) specified in a declarative way.
- This information, i.e.,
  - (language-independent) rules to treat text formats
  - specification of the part-of-speech tagger
  - morphological and bilingual dictionaries and dictionaries of orthographical transformation rules
  - structural transfer rules

  should be provided in an interoperable format $\Rightarrow$ XML.

Rationale /5

Reasons for the development of Apertium as free/open-source software:

- To give everyone free, unlimited access to the best possible machine-translation technologies.
- To establish a modular, documented, open platform for shallow-transfer machine translation and other human language processing tasks.
- To favour the interchange and reuse of existing linguistic data.
- To make integration with other free/open-source technologies easier.

The Apertium platform

Apertium is a free/open-source machine translation platform (`http://www.apertium.org`) providing:

1. A free/open-source, modular, shallow-transfer, language-independent machine translation **engine** with:
   - text format management
   - finite-state lexical processing
   - statistical lexical disambiguation
   - shallow transfer based on finite-state pattern matching
2. Free/open-source **linguistic data** in well-specified XML formats for a variety of language pairs
3. Free/open-source tools: **compilers** to turn linguistic data into a fast and compact form used by the engine and software to learn disambiguation or translation rules.

The Apertium engine/2

Communication between modules: text (Unix "*pipelines*").
Advantages:

- Simplifies diagnosis and debugging
- Allows the modification of data between two modules using, e.g., filters
- Makes it easy to insert alternative modules (interesting for research and development purposes)
- An example: some language pairs have an additional constraint grammar module (based on VISL CG3) before the part-of-speech tagger.

Morphological analyser

- segments the source text in *surface forms* (SFs),
- assigns to each SF one or more *lexical forms* (LFs), each one with:
  - lemma
  - lexical category (part-of-speech)
  - morphological inflection information
- processes contractions (`en`: *can't=can+not*; *won't=will+not*) and multi-word units which may be invariable (`es`: *a través de* [=`en` *through*, *across*]) or variable (`es`: *echó de menos* → *echar de menos* [=`en` *missed*]).
- reads finite-state transducers generated from a morphological dictionary in XML (using a compiler).

Lexical transfer module

- reads each SL LF and generates the corresponding TL LF
- reads finite-state transducers generated from bilingual dictionaries in XML (using a compiler).
- may be invoked before the the structural transfer module or through it

Structural transfer /2

For “harder” language pairs, a three-stage structural transfer is available:

- Patterns of LFs (*chunks*) are detected, processed and marked
- Patterns of *chunks* are detected and processed: this *interchunk* processing allows for longer-range (“inter-chunk”) syntactic transformations
- The output *chunks* are *finished* and the resulting LFs are written.
- In some language pairs, developers have hacked structural transfer with more than three stages.

Post-generator

- Performs some TL orthographical transformations, such as contractions (`ca`: *de* +*els* $\rightarrow$ *dels*; `pt`: *dizer* + *o* $\rightarrow$ *dizê-lo*); `en`: *can* + *not* $\rightarrow$ *cannot*); inserting apostrophes (`ca`: *de* + *amics* $\rightarrow$ *d’amics*), etc.
- It is based on finite-state transducers generated from a post-generation rule dictionary (using a compiler).

Language-pair data

The Apertium project hosts the development of a large number of language pairs:

- *Stable* language pairs include: af↔nl, br→fr, ca→eo, ca↔oc, ca→it, cy→en, es→ast, en↔ca, en↔es, en↔gl, en↔eo, es↔an, es↔ca, es→eo, es↔fr, es↔gl, es↔pt, es↔oc, eu→es, eu→en, fr↔ca, fr→eo, fr↔es, hbs↔slv, kaz↔tat, id↔ms, is→en, is↔sv, mk↔bg, mk→en, mt→ar, nn↔nb, pt↔ca, pt↔gl, ro→es, sh→mk, sme→nob, sv→da
- There is also a growing number of language pairs under development.

The Apertium community

Not the ideal community development situation, but close.

- Very active group of hundreds of developers in sf.net/projects/apertium
- Wiki documentation (wiki.apertium.org)
- IRC channel #apertium in freenode.net
- Mailing lists: apertium-stuff@lists.sf.net and other lists

Research and business with Apertium

Apertium is already an active research and business platform:

- **Research:** 40+ publications, 2 PhD thesis, 4 master's theses
- **Business:** companies (Prompsit, Eleka, Imaxin|software, etc.) offering services to customers such as Autodesk, the Government of Catalonia, one of the main Basque banks, the daily newspaper *La Voz de Galicia*, etc.)

The free/open-source model creates a **community** which effectively connects **researchers**, **developers**, **vendors** and **users**.

These slides are free/open-source