Open-source machine translation: an opportunity for minor languages

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., E-03690 St. Vicent del Raspeig (Spain)

May 23, 2006: 5th SALTMIL workshop, LREC 2006, Genoa, Italy

Contents

Minor languages and minor language pairs/1

What is a minor language? Many alternate denominations are used (in “Google order”):

- *minority* languages (SALT**MIL**)
- *lesser-used* languages (EB**LUL**)
- *minor* languages
- *small* or *smaller* languages
- *lesser* languages
- *under-resourced*, *resource-poor* or *less-resourced* languages
- etc.

Minor languages and minor language pairs/2

What is a minor language?

- Small number of [literate] speakers
- Used far from normality (used more at home that in school or administration, socially discriminated, politically repressed, etc.)
- Lacking a stable writing system, spelling, or standard variety.
- Limited presence on the Internet
- Lacking linguistic expertise
- Lacking machine-readable resources: dictionaries, corpora, etc.

Minor languages and minor language pairs/3

In translation technology, effects on a minor language will occur through language *pairs*.
Example:

- minor languages *A* and *B* are related languages (it is easy to build translation software between them)
- *C* is a major language
- there is translation software from *C* to *A*

Therefore, it will be easier to have translation software from *C* to *B*

Open-source and free software

Open-source software is also called *free* software:

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

For conditions 2 and 4 to be met, anyone should be able to access the source code, hence the name *open source*.

Machine translation software/1

- MT 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:
  - The *engine* (also *decoder*, *recombinator*...)
  - *Data* (linguistic data, corpora)
  - *Tools* to maintain these data and convert them to the format used by the engine

Machine translation software/2

I will focus on RBMT. Reasons:

- CBMT requires massive amounts of sentence-aligned parallel text (hard to come by in minority languages)
- RBMT may use linguistic data elicited by speakers without access to machine-readable corpora
- I am more familiar with RBMT!

MT software/3 : commercial machine translation

- Most commercial MT systems are RBMT (but: LanguageWeaver, Google Labs)
- They use proprietary technologies which are not disclosed (perceived as a main competitive advantage)
- Only partial modification (*customization*) of linguistic data is allowed

MT software/4: open-source machine translation

- For MT to be open-source, engine, data and tools must all be open-source.
- In the case of CBMT this means that corpora must also be open.

MT software/5: MT that is neither commercial nor open-source

There are other possibilities

- Systems that can be freely used over the Internet (some of them not even commercial)
- The engine and tools may be well-documented closed-source programs, with linguistic data being open source (discussed later).

Effects of the availability of MT on minor languages

The availability of MT for a minor language may

- increase its “normality”
- increase its literacy levels
- have an effect on its standardization
- increase its “visibility”

Increasing “normality”

MT may contribute to the normality of a minor language:

- translating educational materials from a major language for schooling in the minor language
- translating news releases from a major language to create minor-language media
- laws, regulations, government informations could be translated to the minor language more easily
- companies would have it easier to market new products in the minor language (“localization”)

[Post-editing of raw MT assumed to be feasible]

Increasing literacy

- Availabilty of text in the minor language (obtained through translation and elaboration) may motivate literacy in the minor language

Effects on standardization

- The existence of a successful MT system may promote:
  - A particular writing system (e.g. Latin without diacritics for Tamazight [=Berber])
  - A particular spelling (Haitian Creole)
  - A particular dialect as a standard (Aranese variety of Occitan)

  by generating linguistic technology for them.

Increasing “visibility”

- Availability of MT from the minor language to major languages may help diffusion of material originally written in the minor language:
  - for instance, MT of websites (on the fly or after postediting)

Commercial MT systems and minor languages: limited opportunities

- MT companies target major world languages (exceptions exist, such as Catalan, but. . . is Catalan really a minor language?)
- It is very hard to adapt closed, commercial MT systems to minor languages

Opportunities from open-source MT systems

- The use of open-source MT systems provides *additional* opportunities, in addition to the *generic* positive effects just discussed:
  - Increases language expertise and resources
  - Increases independence

Increasing expertise and language resources

- When building open-source MT for a minor language, a variety of situations may occur.
- All of them involve building minor-language expertise and resources through
  - reflection about the minor language
  - elicitation of linguistic (monolingual and bilingual) knowledge about it
  - subsequent encoding of this knowledge
- The open-source setting makes new expertise and resources naturally available to the community

Building data for an existing MT engine from scratch

- One needs:
  - A freely available (open-source or not) MT engine
  - Freely available (open-source or not) tools to manage linguistic data
  - Complete documentation on how to build linguistic data for use with the engine and tools
- This is a very unfavourable setting. Many decisions have to be made, e.g:
  - Defining the set of lexical categories and inflection indicators
- The *blank sheet syndrome* may paralyze the project.
- If overcome, the expertise acquired and the resulting open-source data could be improved or used for other purposes: positive effect on the minor language.

Building data for an existing MT engine from existing language-pair data

- If free tools and engine and open-source data are available for another pair with a similar or related language, the *blank sheet syndrome* is drastically reduced. One could, for example:
  - use the same set of lexical categories and inflection indicators
  - build inflection paradigms on top of existing ones

Adapting a new open-source engine or tools for a new language pair

- If source code is available for the engine and tools, experts could enhance or adapt them to address new features of the minor language not dealt with adequately by the current code:
  - character sets
  - structural transfer not powerful enough, etc.
- More challenging than building new data
- But data programmers do not need to have full command of the minor language (abstract management of linguistic issues possible).

Code rewriting would add expertise and resources to the language community.

Increasing independence

- Having an open-source engine, tools and data makes users of the minor language less dependent on a single commercial, closed-source provider
- This has an analogous effect, not only on machine translation, but also on other language technologies.

Standardization of the minor language

Machine translation could accelerate the standardization of a minor language, but this has a downside:

- the lack of a commonly accepted writing system, spelling or reference dialect may pose a serious challenge (“the pioneer syndrome”) to developers.

Neutralizing technophobic attitudes

- For success one must connect well-educated minor-language activism with information-technology literacy
- But this is challenged by `technophobic` attitudes: literate people distrust technologies because of:
  - an idealized view of language and communication
  - a low appreciation for non-formal or non-literary uses
  - a focus on low-probability machine-resistant *jewels* (rather unique words and structures) instead of on high-probability machine-tractable *building blocks* (everyday words and structures)

These “socio-academic” adversities occur (I have experienced them in the Catalan scene).

Organizing community development/1

- Assume we are just developing linguistic data.
- Open-source makes it possible for a minor-language community to collaboratively develop machine translation for it.
- Many languages far from normality have activist groups including people with linguistic and translation skills
- But volunteered time and language and translation skills are necessary but not sufficient.

Organizing community development/2

Some structure is necessary:

- A *coordinating team* mastering the engine and tools used is needed to lead the effort, including:
  - a *code captain* (installing, maintainance, modifications to the code)
  - a *linguistic captain* (linguistic data maintenance)
- A *project web server*
  - to distribute the last version of the system
  - to execute it online
  - for volunteers to contribute new linguistic data
- A group of skilled *volunteers*, certified by the coordinating team.

Eliciting linguistic knowledge

- Existing linguistic knowledge has to made explicit (*elicited*) to contribute it to the system.
- Elicitation of *lexical* knowledge is possible through well-designed web form interfaces:
  - to provide the lemmas of the source and target word
  - to select the inflection paradigm of the source and target word
  - to establish the scope of the equivalence (bidirectional, left-to-right, right-to-left).
- Elicitation of other knowledge (e.g., structural transfer rules) is harder(a subject of research indeed).

Simplicity of linguistic knowledge needed

The level of linguistic knowledge necessary to start build a new MT system should be kept to a minimum (basic high-school grammar skills and concepts).

- This is rather easy in *shallow-transfer* MT systems.
- But is very difficult (if not impossible) for *deep transfer* systems.

Well-written documentation may be very helpful.

Standardization and documentation of linguistic data formats

- An adequate documentation of the format of linguistic data is crucial.
- The way: using XML. Why?
  - Each data item is *explicitly* labeled with a descriptive, named tag with a clear meaning attached
  - The structure of documents may easily be validated against DTDs or schemas
  - Many technologies exist for XML (converting from and to XML, *interoperability*).

Modularity

- The emphasis of open-source is the reusability of code and linguistic data to build new MT systems or other language-technology applications.
- For that objective *modularity* is a must.
- A modular engine induces modularity in its data.
- For example, having an independent morphological analyser and an independent morphological dictionary
  - Makes it easier to build an MT system for a different target language
  - May be used to build an intelligent search engine (inflection-independent search)

Case study: Opentrad Apertium and Aranese/1

[Details: Armentano-Oller & Forcada (poster)]

- Apertium is an open-source shallow-transfer MT toolbox
- It is adequate for MT between related languages
- Comes with open-source data for `es`–`ca`, `es`–`gl`, `es`–`pt`[1]
- Being currently developed through the SourceForge platform
- Developers sought (`www.apertium.org`)

[1] `es`=Spanish, `ca`=Catalan, `gl`=Galician, `pt`=Portuguese

Case study: Opentrad Apertium and Aranese/2

- A new pair being developed, between:
  - a *small* language (Catalan, $\simeq$6.000.000 speakers) and
  - a *very small* language (Aranese, $\simeq$6.000 speakers, a *standardised* dialect of Occitan, oc, $\simeq$1.000.000 speakers?)
- Preliminary results in $\simeq$*2 person-months* (oc–ca)
  - Text coverage (words known): 89%
  - Error rate: 9%

Case study: Opentrad Apertium and Aranese/3

When a bidirectional 98%-coverage and 6%-error system, e.g., is available (Dec. 2006):

- The amount of Aranese text in the web may increase (visibility)
- The weight of the Aranese dialect in the ongoing Occitan standardisation may increase
- The wider Occitan community (mainly in France) may modify it for, say Occitan–French
- Aranese data will be available for other language-technology applications

Concluding remarks

After reflecting on the matter, I can (tentatively) conclude:

- MT can have a positive effect on minor languages (normality, “visibility”, literacy, standardization)
- *Open-source MT* can have *specific*, *additional* effects (increasing expertise, contributing reusable resources, reducing technological dependency).
- Development of MT for minor languages faces a number of challenges: (lack of standardization, technophobic attitudes, elicitation of linguistic knowledge, need for standard formats, modularity).

Of course, I will be happy to discuss these conclusions!

Acknowledgements

- Partial funding from:
  - Ministry of Science and Technology (grant TIC2003-08681-C02-01)
  - Ministry of Industry, Tourism and Commerce (grants FIT-340101-2004-3 and FIT-340001-2005-2).
- I thank A.M. Corbí-Bellot, M. Ginestí-Rosell, J.A. Pérez-Ortiz, G. Ramírez-Sánchez, F. Sánchez-Martínez, S. Ortiz–Rojas, C. Armentano–Oller and M.A. Scalco for comments, suggestions.
- And I thank the organizers for inviting me to this Workshop!