

OPEN TRANSLATION TOOLS

Published : 2011-03-12

License : None

INTRODUCTION

1. Introduction
2. About This Manual

1. INTRODUCTION

The first wave of the internet revolution changed expectations about the availability of information a great deal. Information that was stored in libraries, locked in government vaults or available only to subscribers suddenly became accessible to anyone with an internet connection. A second wave has changed expectations about *who* creates information online. Tens of millions of people are contributing content to the modern internet, publishing photos, videos, and blog posts to a global audience.

The globalization of the internet has brought connectivity to almost 1.6 billion people. The internet that results from globalization and user-authorship is profoundly polyglot. Wikipedia is now available in more than 210 languages, which implies that there are communities capable of authoring content in those tongues. Weblog search engine Technorati sees at least as many blog posts in Japanese as in English, and some scholars speculate that there may be as much Chinese content created on sites like Sina and QQ as on all English-language blogs combined.

A user who joins the internet today is far more likely to encounter content in her own language than had she joined ten years ago. But each internet user is able to participate in a smaller percentage of the total interactions and conversations than an English-speaking internet user could have in 1997, when English was the dominant language of the Net.

There's a danger of linguistic isolation in today's internet. In an earlier, English-dominated internet, users were often forced to cross linguistic barriers and interact in a common language to share ideas with a wider audience. In today's internet, there's more opportunity for Portuguese, Chinese, or Arabic speakers to interact with one another, and perhaps less incentive to interact with speakers of other languages. This in turn may fulfill some of the predictions put forth by those who see the internet acting as an echo chamber for like-minded voices, not as a powerful tool to encourage interaction and understanding across barriers of nation, language and culture.

For the the internet to fulfill its most ambitious promises, we need to recognize translation as one of the core challenges to an open, shared, and collectively governed internet. Many of us share a vision of the internet as a place where the good ideas of any person in any country can influence thought and opinion around the world. This vision can only be realized if we accept the challenge of a polyglot internet and build tools and systems to bridge and translate between the hundreds of languages represented online.

Machine translation will not solve all our problems. While machine translation systems continue to improve, they are well below the quality threshold necessary to enable readers to participate in conversations and debates with speakers of other languages. The best machine translation systems still have difficulty with colloquial and informal language, and are most reliable in translating between romance languages. The dream of a system that creates fully automated, high quality translations in important language pairs like English-Hindi still appears long off.

While there is profound need to continue improving machine translation, we also need to focus on enabling and empowering human translators. Professional translation continues to be the gold standard for the translation of critical documents. But this method is too expensive to be used by web surfers simply interested in participating in discussions with peers in China or Colombia.

The polyglot internet demands that we explore the possibility and power of distributed human translation. Hundreds of millions of internet users speak multiple languages; some percentage of these users are capable of translating between these. These users could be the backbone of a powerful, distributed peer production system able to tackle the audacious task of translating the internet.

We are at the very early stages of the emergence of a new model for translation of online content -- "peer production" models of translation. Yochai Benkler uses the term "peer production" to describe new ways of organizing collaborative projects beyond such conventional arrangements as corporate firms. Individuals have a variety of motives for participation in translation projects, sometimes motivated by an explicit interest in building intercultural bridges, sometimes by fiscal reward or personal pride. In the same way that open source software is built by programmers fueled both by personal passion and by support from multinational corporations, we need a model for peer-produced translation that enables multiple actors and motivations.

To translate the internet, we need both tools and communities. Open source translation memories will allow translators to share work with collaborators around the world; translation marketplaces will let translators and readers find each other through a system like Mechanical Turk, enhanced with reputation metrics; browser tools will let readers seamlessly translate pages into the highest-quality version available and request future human translations. Making these tools useful requires building large, passionate communities committed to bridging a polyglot web, preserving smaller languages, and making tools and knowledge accessible to a global audience.

2. ABOUT THIS MANUAL

This manual was collaboratively designed and written by a community of Open Translation innovators using the FLOSSManuals platform. It is the product of the first-ever Open Translation Tools Book Sprint, and builds on work done at two Open Translation Tools convergences, a pair of live events designed by Aspiration (www.aspirationtech.org), and realized in collaboration with a wonderful set of partner organizations and the support of generous and forward-looking funders.

The Open Translation Tools Book Sprint was held in De Waag, a beautiful historic building located in the center of Amsterdam, kindly provided as a venue by De Waag Society for Old and New Media (www.waag.org). Many thanks to Lucas Evers and Christine van den Horn for organising the venue and being fantastic hosts.



The first Open Translation Tools Convergence (OTT07) took place in late 2007 in Zagreb, Croatia, co-organized by Aspiration and Multimedia Institute (www.mi2.hr). Supported by the generosity of the Open Society Institute (www.soros.org), with additional support provided by TechSoup Global (www.techsoupglobal.org), this event produced the initial framing paper on Open Translation, www.aspirationtech.org/paper/opentranslationtools.

The second Open Translation Tools event was held in Amsterdam in June 2009, and was co-organised by Aspiration, FLOSS Manuals (www.flossmanuals.net), and Translate.org.za. OTT09 was again supported by the Open Society Institute, with generous additional travel support from the Ford Foundation (www.fordfound.org). OTT09 was held at Theater de Cameleon (www.decameleon.nl), who provided a stunning facility and top-notch hospitality.



Both OTT events ran for three days, and were attended by a total of more than 140 people from over 40 different countries, speaking over 50 different languages.

The OTT agendas were collaboratively developed by participants and event organizers before and during the gatherings, and the proceedings were directed using Aspiration's collaborative approach to event facilitation (facilitation.aspirationtech.org). Each session was run as a discussion led by one of the participants. All sessions were documented with notes that can be found on the OTT wiki (ott09.aspirationtech.org).

Throughout the OTT09 conference, participants were invited to contribute to the proposed index for the Open Translation Tools book and to learn the FLOSS Manuals tool set so they could contribute remotely.



The Open Translation Tools Book Sprint immediately followed OTT09 at De Waag. Directed by Adam Hyde of FLOSS Manuals, over a dozen participants worked from 10.00 to 22.00 each day on the book, iteratively developing content and grouping chapters while discussing terminology, technology, licensing, and a wealth of other Open Translation topics.

The manual was written in five days but the maintenance of the manual is an ongoing process to which you may wish to contribute.

HOW TO CONTRIBUTE TO THIS MANUAL

To contribute, follow these steps:

1. Register

Register at FLOSS Manuals:

<http://en.flossmanuals.net/register>

2. Contribute!

Select the manual <http://en.flossmanuals.net/bin/view/OpenTranslationTools> and a chapter to work on.

If you have questions about how to contribute, join the chat room listed below and ask! We look forward to your contribution!

For more information on using FLOSS Manuals, read our manual:

<http://en.flossmanuals.net/FLOSSManuals>

CHAT

It's a good idea to talk with us so we can help co-ordinate all contributions. A chat room is embedded in the FLOSS Manuals website so you can use it in the browser.

If you know how to use IRC you can connect to the following:

server: `irc.freenode.net`

channel: `#flossmanuals`

MAILING LIST

For discussing all things about FLOSS Manuals, join our mailing list:

<http://lists.flossmanuals.net/listinfo.cgi/discuss-flossmanuals.net>

ABOUT THE AUTHORS

This manual exists as a dynamic document on flossmanuals.net, and over time will have an ever-increasing pool of authors and contributors.

The following individuals were part of the 2009 Open Translation Tools Book Sprint. We thank them for their tireless efforts to create this first-of-its-kind volume.

Adam Hyde, FLOSS Manuals

Ahrash Bissell, Creative Commons

Allen Gunn, Aspiration

Anders Pedersen

Andrew Nicholson, Engage Media

Ariel Glenn, Wikimedia

Ben Akoh, Open Society Initiative for West Africa

Brian McConnell, Worldwide Lexicon

David Sasaki, Global Voices Online

Dwayne Bailey, translate.org.za

Ed Bice, Meedan

Ed Zad, dotSUB

Edward Cherlin, Earth Treasury

Ethan Zuckerman, Berkman Center for Internet and Society

Eva-Maria Leitner, University of Vienna

Francis Tyers, Universitat d'Alacant

Georgia Popplewell, Global Voices Online

Gerard Meijssen, Stichting Open Progress

Javier Sola, WordForge Foundation

Jeremy Clarke, Global Voices Online

Laura Welcher, dotSub and Global Lives

Lena Zuniga, Sula Batsu

Matt Garcia, Aspiration

Mick Fuzz, Clearer Channel

Mike Roman, Aspiration

Patrice Riemens

Philippe Lacour, Zanchin

Sabine Cretella, Anaphraseus

Silvia Florez, Universitat Jaume I

Thom Hastings, City Year

Thomas Middleton

Wynand Winterbach, translate.org.za

Yves Savourel

ACKNOWLEDGMENTS

This manual is a culmination of almost three years of research, planning, convening, and collaboration.

Aspiration first proposed a program in Open Translation to the Open Society Institute (OSI) in 2006. OSI subsequently funded two Open Translation Tools convergences, in Zagreb in 2007 (OTT07) and in Amsterdam in 2009 (OTT09), as well as the Open Translation Tools Book Sprint following OTT09. The Ford Foundation and TechSoup Global also provided generous travel support for event participants. We are deeply grateful to all our funders for their generous and forward-looking support.

Aspiration would like to formally thank the following individuals and organizations:

Contributors to the Open Translation Tools Book Sprint, who worked tirelessly over five days to create a first-of-its-kind volume on Open Translation.

All the **participants** and **facilitators** at OTT07 and OTT09, whose shared wisdom and knowledge are aggregated in these pages. In particular, thanks to those who took notes during sessions for the wiki, as that material forms the basis for substantial parts of this document, and to those who contributed ideas towards the design of the book.

FLOSS Manuals (www.flossmanuals.net) and **Adam Hyde**, who co-organized OTT09 and directed the Book Sprint that generated this volume. We salute FLOSS Manuals's vision and leadership in the field of free and open documentation, and the innovative platform they have developed.

Translate.org.za (translate.org.za) and **Dwayne Bailey**, who co-organized OTT09 and whose leadership in the fields of FLOSS translation and localization is unparalleled.

Tomas Krag, who pioneered the Book Sprint concept with the creation of *Wireless Networking in the Developing World* (www.wndw.net).

De Waag Society for Old and New Media (www.waag.org) and **Lucas Evers** and **Christine van den Horn**, who provided an amazing venue for the Book Sprint, fantastic hospitality, and also organized the book publication reception.

Theater de Cameleon (www.decameleon.nl), who provided a stunning facility and top-notch hospitality for OTT09.

Ethan Zuckerman (www.ethanzuckerman.com), who has been a tireless champion of Open Translation in his work with [Global Voices Online](http://GlobalVoices.org) and elsewhere, and who contributed his essay "[The Polyglot Internet](#)" for the introduction of the Open Translation Tools manual.

Hotel Van Onna (www.hotelvanonna.nl), which provided wonderful accommodations and hospitality for the OTT09 Book Sprint participants in Amsterdam's Jordaan neighborhood.

Multimedia Institute of Zagreb (www.mi2.hr), who co-organized the OTT07 event that started all the fun, serving as passionate participants and collaborative partners without equal. OTT07 simply would not have been possible without their leadership and support, and the high quality of participant experiences there was a direct result of their exhaustive attention to detail and hospitality.

Open Society Institute (www.soros.org), which provided the funding to make OTT07, OTT09 and the Open Translation Tools Book Sprint possible, and **Janet Haven**, whose guidance and support in the development of Aspiration's program in Open Translation have been ongoing.

Ford Foundation (www.fordfound.org), which provided support for travel to OTT09 that allowed key participants to join in the proceedings.

TechSoup Global (www.techsoupglobal.org), which provided support for travel to OTT07 that allowed key participants to join in the proceedings.

In short, we thank everyone who has been involved in the Open Translation program to date, and we hope to find many opportunities to meet together again and further strengthen this nascent network of practice.

OPEN TRANSLATION

3. Why Translate?

4. The Vision for Open Translation

5. The State of Open Translation Tools

3. WHY TRANSLATE?

A foundational question in explaining and advocating Open Translation is "why translate?". Though reasons vary, there is a range of inspirations and mandates for translation and broader access to content.

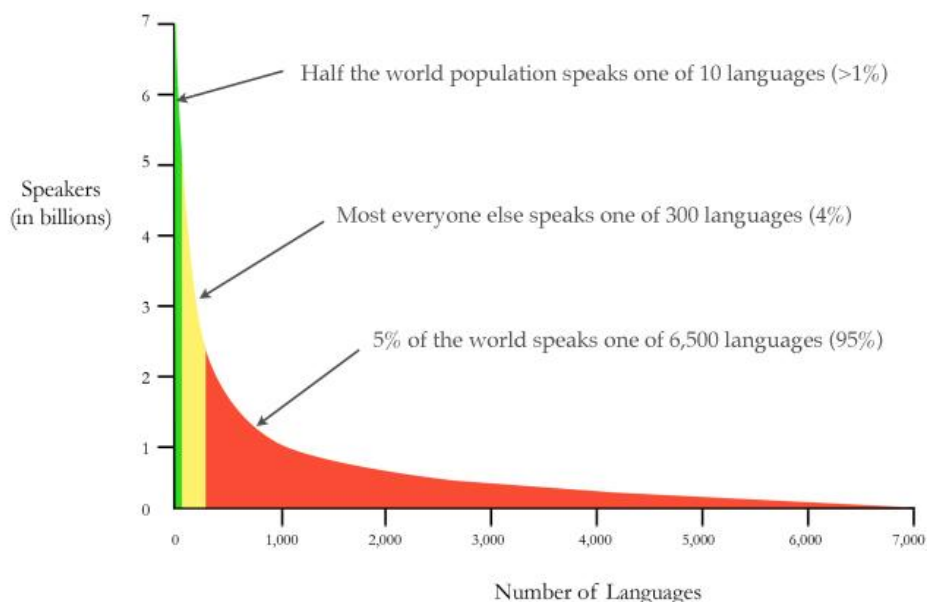
MOTIVATIONS FOR TRANSLATING

For those in the open knowledge and open education fields, translation is about access to information and associated issues of social justice. Until all peoples have access to the most up-to-date and complete knowledge, a globally equitable world is not possible.

For those in the FLOSS (Free/Libre Open Source Software) universe, key motivators are expanding access to documentation and training resources, as well as creating localized versions of software tools. A framing value of the Free Software movement is the right of individuals to modify software to meet their specific needs, and language-specific versions of software certainly exemplify this freedom.

Other motivations to translate are:

- *Creating new audiences and markets* -- Making content available in additional languages expands the potential pool of consumers for that material. For instance, making an article available in Chinese (and getting it through the Great Firewall) invites potential attention from an audience that numbers in the billions.
- *Promoting mother tongues* -- In a related dynamic, translation allows communities to promote their mother tongue, both by translating relevant new information into the language, as well as publishing information available only in that language into other languages. Translation into just a few widespread languages, like Hindi, Spanish, Arabic, Russian, and Japanese, can reach about half the people on the planet. Enabling translation into about 300 additional languages could reach just about everybody else (<http://www.ethnologue.com>). With cell phone technology expanding around the globe, many millions are poised to come online, driving demand for content and communications in their native tongues.



- *Revitalizing endangered languages* -- UNESCO (United Nations Educational, Scientific and Cultural Organization) estimates that nearly half of the approximately 6,000 languages spoken on earth are endangered, and many more are in a precarious state (<http://www.unesco.org/culture/ich/index.php?pg=00206>). With the loss of a language comes incalculable cultural loss, as well as loss to all humanity of our knowledge about the planet and how to live sustainably in its myriad environments. Enabling translation into endangered languages empowers speakers and adds vitality to their languages by allowing them a voice in a new, modern online domain. Translation in both directions generates language-pair corpora that support subsequent translations, and new tools powered by machine analysis.
- *Cross-pollinating ideas* -- Translated content provides perspective and knowledge from different cultures, experiences, and intellectual frames. Even in the internet era, much information still resides in language-specific silos. Making such resources available to broader audiences propagates ideas and drives the consideration of different schools of thought and belief.
- *Stimulating local content creation* -- Translated content can be a catalyst for additions, responses, and new creation in the local language. As YeeYan (www.yeeyan.com) translates content into Chinese for publication within China, they are engendering a new generation of Chinese content producers, both those who translate and those who comment and collaborate around the content itself.
- *Aligning with national objectives or legislation* -- In Canada, products must be released in both French and English, and similar situations exist in many other countries. Sometimes it's not just a good idea, it's the law.
- *Matching content with community language needs* -- There are specific efforts in South Africa aimed at producing HIV/AIDS content for impacted communities. In such situations, translation is a life-saving and community-sustaining service.

THE BROADER CASE FOR TRANSLATION

In general terms, translation enables us to understand our fellow global citizens. The ongoing globalization of knowledge, communication, and information networks has the potential to be a hopeful story. There is vast untapped potential for diverse collaborations between the 1.8 billion users who are on line -- the potential to share knowledge and converse through an increasingly capable infrastructure of open translation tools, open translation data, and online translation communities. There are also many examples of a lack of translation foiling the best intentions. For example, the name of the online invitation and social event site, Evite, tells Spanish-speaking users, "avoid". Translation is crucial for a global society to function together.

When we think about the role of translation in transforming the global web, we can consider several major areas of impact: media, education, health resources, and software.

The media environment is changing. Increasingly, the way we communicate with the people around us is an act of publication. Every moment there is a voice in the world that most needs to be understood, whose poetic insight or dramatic experience or unique knowledge is potentially world-changing. So-called "citizen journalism" offers real-time and real-place access to understanding our world. Translation is the missing ingredient in a participatory global media ecosystem that could lead to a world with a more complete and more nuanced understanding of the events that shape our shared circumstances. This becomes all the more true as dialog scales about our responses to climate change and other global challenges.

Open Educational Resources (OER) offer an opportunity to scale learning across the globe. With ever more learning material available under open licenses and with increasing global internet penetration, the primary remaining barrier to making this material accessible is translation. In one early proof of concept, Meedan.net implemented a pilot project for Teachers Without Borders that combined machine translation with crowd-sourced (social) error correction on top of OER for Arab region secondary school learners.

As the world becomes increasingly connected and interdependent, diseases and health concerns move faster, while continuing to ignore borders and language barriers. Global health efforts against illnesses such as HIV/AIDS, tuberculosis, and malaria depend on successful transmission of accurate prevention and care information. This can only occur with proper translation. Health information is an area where translation is essential to saving lives.

While participatory media and open knowledge networks offer global citizens better access to content, the free software movement enables emergent knowledge economies around the world. Projects like translate.org.za empower language communities with tools to localize open source software into local languages.

BUT WHAT ABOUT MY CONTENT?

Why translate your website or blog? There are many reasons to consider publishing a multilingual website, among them reaching local readers who speak other languages, engaging international readership, and increasing search engine visibility for your content in other languages.

Local Readership

Most cities are multi-ethnic and multi-lingual. Even in the United States, which many people consider an English speaking country, a sizable minority of the population speaks English as a second language or not at all. In most major US cities, the Spanish-speaking audience alone is significant. If you publish a local or regional website, for example an online newspaper or local events blog, you should consider targeting the most important secondary languages in your market, which might otherwise never hear your message.

International Readership

If your website has an international audience (this is easy to see with services like Google Analytics) or covers a topic that is not tied to a region, you can expand your readership and visibility by targeting important international languages (for example, English, Spanish, French, or Chinese). Once your content is routinely translated to these languages, you'll become visible and linkable in these languages and countries, and should start receiving traffic from these regions that you would otherwise never have received.

Search Engine Visibility

People search for sites or terms in their language, not yours. Translating your site into other languages makes your site more visible to search engines, which will index your site and relevant search terms in those languages. You will soon become visible to people doing keyword searches on those terms, and to other websites, which may begin to link to you as well. Blogs are an especially important source of "side door" links in other languages, and as more of them link to you, your search engine ranking will improve.

TRANSLATION IS A LOCAL DECISION

In the end, each creator of content makes his or her own decision about its readiness and availability for translation. This book is published on the dual premises that translation is an imperative for better global understanding in an increasingly complicated world, and that open translation is the most appropriate, scalable and sustainable approach to making content and knowledge available to the broadest set of communities and citizens.

4. THE VISION FOR OPEN TRANSLATION

Open Translation describes a nascent field of practice emerging at the crossroads of three dynamic movements of the information and internet eras:

- Open Content
- Free/Libre/Open Source Software (FLOSS)
- Open/Peer Production

Open Content encompasses a diverse range of knowledge resources available under open licenses such as Creative Commons (CC) and Free Document License (FDL), from books to manuals to documents to blog posts to multimedia. These resources are published on terms that encourage their redistribution, modification, and broad re-use. Open content resources like Wikipedia have dramatically changed the way knowledge is authored, maintained, and accessed.

The **Free/Libre/Open Source Software (FLOSS)** movement is a vibrant global phenomenon which has, over the past 30-plus years, generated a sprawling ecology of software tools that are freely and openly available to anyone who wants them. This movement has established an alternative to proprietary, corporate-controlled software and corresponding closed data formats, which greatly benefits translators and localization practitioners. Given access to the underlying source code, they can create new versions of tools to support underserved audiences.

Open- or peer-production models use the internet's connected-but-distributed nature to bring broad human resources to bear on specific tasks or problems. Wikipedia is the flagship example of this, existing today as the single largest knowledge set in history. Other examples of peer production include Project Gutenberg's distributed proofreading community (www.gutenberg.org) and the FLOSSManuals authoring platform (www.flossmanuals.net) on which this book is produced.

Open Translation synthesizes these three models of open production and open collaboration into a new discipline. It is the set of practices and work processes for translating and maintaining open content using FLOSS tools, and using the the internet to make that content and those tools and processes available to the largest number of writers and readers. Open Translation tools comprise a body of software that supports or performs language translation and is distributed under a FLOSS license.

Open Translation's open components are fundamentally interrelated.

- If translation of open content depends on non-free or non-open software, it creates a critical bottleneck in the open knowledge ecosystem. When translation access and flow are controlled by proprietary tool vendors, those vendors can charge high prices and can disregard the best interests of open publishers and their readers.
- Because FLOSS projects are open source, FLOSS translation tools can always be further localized and customized to support new language pairs and locales.
- For free and open source software, open content is appropriate. Using FLOSS tools to translate and manage non-open content is prevalent and mostly license-compliant (with the GNU General Public License version 3 a notable exception). Free and open tools should ideally operate on free and open data.
- The open production models of Open Translation lower the barriers to participation in cross-language knowledge exchange, and help avoid replication of the "expert culture" that permeates the professional translation industry.

Open translation can be viewed as translation's movement from an individual sport to a team sport. Additionally, social translation on the Internet is, as Ethan Zuckerman has suggested, a way for communities of translators to become journalists, deciding which content to move between language communities. Journalism on the web, as a social practice, is as much about curating, annotating, rating, and linking as it is about writing. This is a powerful and emergent form of journalism, encyclopedia creation, social networking, and much more.

THE PROMISE OF OPEN TRANSLATION

Open content projects like Wikipedia have rewritten conventional wisdom on who can publish knowledge. Global Voices Online has dramatically prefixed the role of 'journalist' with the adjective 'citizen'. The Free and Open Source software movements have inverted software production models from centralized, opaque and often lurching processes into decentralized, transparent and frequently agile endeavors.

Open Translation promises to profoundly broaden access to knowledge across language barriers. Wikipedia may exist in hundreds of languages, but many language versions lag in terms of coverage. General cross-lingual access to open content and digital knowledge is still the exception rather than the rule. The future of Open Translation lies in establishing richer, better-connected sets of online and offline tools while growing a global network of volunteer translators who understand and follow best practices for translating content and building open translation memories.

The vision for Open Translation is predicated on the notion that anyone can be a translator by contributing to Open Translation projects. Just as FLOSS projects have project managers, testers, community moderators, and documenters in addition to developers, Open Translation projects welcome the efforts of proofreaders, editors, and project managers in addition to actual translators.

But there are also opportunity costs to adhering to a vision of Open Translation. Open Translation tools are in many instances not as mature or full-featured as their proprietary counterparts. Those wishing to blaze the trail of an all-open approach to translation face a worse-before-better situation, where near-term sacrifice is necessary to support the improvement and evolution of the open tool set.

GETTING TO OPEN TRANSLATION

As the field of Open Translation continues to emerge and evolve, there are a number of projects, networks, issues and trends driving and gating that evolution.

As much as open content, FLOSS, and peer production models have profoundly impacted our world and culture, they are not yet well integrated for the purposes of Open Translation. Open Translation tool coverage is incomplete, and existing tools rarely inter-operate or share standards for data interchange. Ubiquitous web publishing platforms like Drupal and Wordpress have minimal built-in support for maintaining multi-lingual sites. Add to these facts that open content license publishers like Creative Commons have not fully resolved licensing implications for translated works, and it is clear there is still much work to be done.

Open Translation is synonymous with a new ecology of participation, and one for which the roles are still being established. What is known is that there are two under-tapped human resources which can be brought to bear: translators who want to volunteer their skills, and poly-lingual individuals who want to serve as volunteer translators. But leveraging such contributions is dependent on having well-defined ways in which to get involved. Global Voices and Wikipedia have fundamentally different models for volunteer translation, and are still evolving their community processes. Most open content projects have no idea how to establish sustainable volunteer translation models, and many that do utilize rudimentary processes centered on exchanging large email attachments.

Scaling the pool of volunteer translators is its own challenge. Bi-lingual abilities are but a pre-requisite to being an effective translator; practice and experiential learning are required to effectively translate. Establishing community hubs for open translators is also an unsolved problem; while several professional translation communities such as ProZ thrive on the internet, open translator communities are only now beginning to gain momentum, and individuals who translate for open content projects are usually operating in project-specific networks.

A fundamental challenge in an open environment is quality control. Traditional, centralized translation models have dedicated editors and proofreaders whose job it is to verify accuracy and consistency of translation. It remains for the Open Translation movement to establish quality processes and transparent mechanisms for reputation measurement.

Also, the significance of regional and cultural issues in translation work can not be overstated; as norms and values vary, a range of secondary connotations and associations must be considered in crafting appropriate translations. While professional translators spend years learning the nuances of idiom and linguistic mapping in specific language pairs, open translators will not as often have the benefit of such learning curves. Creating better open repositories of essential empirical knowledge and best practices will further accelerate the ramp-up of volunteer translators and the quality of translations.

THE VISION TURNS ON WHEN, NOT WHETHER

An openly translated internet is ultimately a matter of time, and the great unknown is how long it will take to realize the vision. Open Translation will scale in proportion to the open tools and open content upon which it rests, and on the corresponding efforts of those leading the way.

This book exists as a step along the path, an attempt to both capture essential knowledge and take measure of the tools, processes and learnings of Open Translation to date. As you read on, consider yourself a part of this movement by virtue of your very interest. We invite you to contribute to the Open Translation movement in any way that taps your passion and inspires your participation.

5. THE STATE OF OPEN TRANSLATION TOOLS

The field of translation is in a state of transition, and software tools to support language translation are evolving with corresponding rapidity. Increasingly available online resources are quickly expanding the possible and the practical when it comes to translating content, and processes and business models which have remained relatively staid for decades are being rethought.

Even in the so-called “broadband” era where substantial parts of the globe enjoy ubiquitous high-speed access and where translation is thus more important than ever, most translators and translation firms have employed rather rudimentary technology processes in their translation workflow. Translators generally copy and paste text between word processor documents and transmit translated documents as email attachments that lack all but the most basic version control or metadata. However, new online tools and innovative new workflow models are turning the translation field on its head.

OPEN TRANSLATION TOOLS TODAY

The state of Open Translation tool offerings reflects the same flux. Real-time access to a global network of translation services and talent is a resource that the translation industry is only now starting to leverage and upstart multilingual projects on the internet are pushing the state of the art by treating translation as an exercise in distributed problem solving.

In addition, most Open Translation tools have recently begun to incorporate workflow, user role tracking, permissions and detailed state information for each translation project. From the RSS-enabled platforms like Worldwide Lexicon, which automate translation requests and submissions, to crowd-sourced tools like dotSUB (which although not open source) employ an open approach to data and translation for subtitling digital videos, Open Translation tools are demonstrating their ability to not only track but also outpace closed and proprietary offerings.

Open Translation tools, then, fall into a range of categories:

- **PO and XLIFF localization editors:** This encompasses offline, online and distributed localization tools that read and write data in PO, XLIFF and related formats. These serve as the essential tools for many translators and localizers. Examples of these tools include Pootle, Poedit, gtranslator, Transolution, and Lokalize.
- **Translation workflow:** These tools manage roles, tasks and other project information, and often interoperate with other translation tools and version control systems. Workflow is a critical area for the growth of Open Translation, and there exists a range of un-met needs in terms of workflow support. Examples of these tools include Transifex, Translate Toolkit, Pootle, Launchpad Translations and Worldwide Lexicon.
- **Subtitling:** As video becomes a more pervasive web offering, tools for adding translated subtitles to videos are becoming more in demand. Examples of such tools include GNOME Subtitles and DotSub.
- **Machine translation:** These tools, which at present are primarily hosted as web sites like translate.google.com and BabelFish, perform algorithmic translation of text from one language to another. Examples of these tools include Apertium and Moses.
- **Translation Memory:** These Computer Aided Translation (CAT) tools store small discrete language fragments, passages, and terms in order to assist human translators as they perform their work. Examples of these tools include QT Linguist and OmegaT.
- **Dictionary and Glossary:** As their names imply, these CAT tools store definitions for terms in a given language, and support translators as they map from one language to another. Examples of these tools include CollaboDict and Transolution.
- **Wiki translation:** These modules and extensions enhance and augment existing wiki platforms with tools for performing and managing translation of wiki content. Examples of these tools include Cross-Lingual Wiki Engine and translation.wik.net.

As with almost any collection of software tools, these categories blur and overlap on a tool-by-tool basis; the categories are somewhat arbitrary and many tools fall into more than one.

A detailed listing of Open Translation tools is available at <http://socialsourcecommons.org/toolbox/show/110>. We encourage readers to add missing tools to that list.

Related tools and resources

There are number of related tool categories and resources which are worth mentioning in the context of Open Translation:

- **Code libraries and packages:** While the focus of this book is on tools for end users in various translation workflows, code libraries are an essential and core element of the Open Translation ecology. Most ubiquitous among the libraries is `gettext`, the API used by a wide range of localization and translation tools to read and write PO files and other translation-related data.
- **Content Management Systems (CMS):** FLOSS CMS platforms offer a range of multilingual capabilities. While no current CMS readily supports a true multilingual web site (that is, either a single site available in multiple languages, or alternately a site on which separate pages can contain text in multiple scripts), many CMS platforms offer good support for translating site content. These include Drupal, Plone, Joomla!, Twiki, and FLOSS Manuals.
- **Operating systems:** End-user support for multi-lingual operating systems is very much the exception; users of Windows, Macintosh, and most Linux distributions install for a given locale, and must often reboot to properly run in a different locale. A noteworthy variant in this regard is `Linguas OS`, a distribution of GNU/Linux operating system adapted for professional translators and those working in software localization.
- **Guides and online resources:** While too numerous to enumerate here, a number of guides and online resources are available to those working in Open Translation. Several of the most noteworthy include the UNDP Localization Primer, LISA publications which provide best practices and primers from Localization Industry Standards Organization, and the wiki at translate.sourceforge.net. A resource specific to GNOME is `Damned Lies`, which is a hub for translation workflow for the GNOME project.

OPEN TRANSLATION FEATURE GAPS

Open Translation is an emergent field and a primary point of discussion is about the areas in which Open Translation tools are lacking. While a range of gaps exist, there were two primary functionality holes that arguably overshadow the rest:

- **Workflow support:** Though a number of Open Translation tools provide limited support for translation workflow processes, there is currently no tool or platform with rich and general support for managing and tracking a broad range of translation tasks and workflows. The internet has made possible a plethora of different collaborative models to support translation processes. But open source tools to manage those processes, tracking assets and state, role and assignments, progress and issues, are few. While tools like Transifex provide support for specific workflows in specific communities, generalized translation workflow tools are still few in number. An ideal Open Translation tool would understand the range of roles played in translation projects, and provide appropriate features and views for users in each role. As of this writing, most Open Translation tools at best provide workflow support for the single type of user which that tool targets.
- **Distributed translation with memory aggregation:** As translation and localization evolve to more online-centric models, there is still a dearth of tools which leverage the distributed nature of the internet and offer remote translators the ability to contribute translations to sites of their choosing which request the same. As of this writing, Worldwide Lexicon is the most advanced platform in this regard, providing the ability for blogs and other open content sites to integrate distributed translation features into their interfaces. In addition, there needs to be a richer and more pervasive capture model for content translated through such distributed models, in order to aggregate comprehensive translation memories in a range of language pairs.

Other Open Translation technology gaps include:

Interoperability

Lack of integration and interoperability between tools means both frustration for users and feature duplication by developers. Different communities have their own toolsets, but it is difficult for a translation project to make coherent use of a complete tool set. Among the interoperability issues which require further attention in the Open Translation tools ecology:

- Common programming interfaces for tools to connect, share data and requests, and collect translation memories and other valuable data.
- Plugins for content management systems to export content into PO-files, so that content can be translated by the wealth of tools that offer PO support.
- Better integration between different projects, including shared glossaries, common user interfaces and subsystems, and rich file import/export.
- Generic code libraries for common feature requirements. "gettext" stands out as one of the most ubiquitous programming interfaces in the Open Translation arena, but many more interfaces and services could be defined and adopted to maximize interoperability of both code and data.

Reviewer Tools

Tools for content review are lacking; features for quality review should be focused on distributed process and community-based translation. As such reviews can be a delicate matter, the ideal communication model when there are quality problems is to contact the translator, but timing can be an issue. In systems with live posts and rapid translation turnaround, quick review is important and it may not be possible to reconnect with the content translator in a timely fashion.

A FUTURE VISION OF OPEN TRANSLATION TOOLS

One of the goals of this book is to drive discussion and creation of better Open Translation tools. This section describes an idealized feature set for the Open Translation tool space, specifying functionality for a tool which does not yet exist, but which would meet the broadest range of text translation needs in terms of features, supported workflows, and business models.

It is important to note that is a purely theoretical exercise; it is generally agreed that large monolithic tools are not the right course for the future, and that a small, distributed set of tools that work well together is the recommended path for better supporting Open Translation efforts.

That said, the described feature set is both expansive and impressive in its ambition to meet a wealth of Open Translation needs. The following sections describe those desired features, grouped into three sets: core features, workflow support, and additional features.

While most of these capabilities are available in various proprietary and open source tools, there is not currently a FLOSS tool or tool set that comes close to offering the features enumerated below.

Core Features

The following should be considered requisite for any idealized functionality. These are primarily features associated with the translation of a single text source; higher-level features are described in subsequent sections.

The following should all be available in the user interface for the tool:

- Original text display would show the source text, using color and iconography to denote progress, commentary and other relevant metadata.
- Output/preview display would render the translated text, maintaining layout from the original and supporting detailed linkage between the source and translated versions of the text.
- A commenting/annotation feature would allow users to select and annotate text in both the source and translated text in order to add comments and other useful annotations to the core data.
- Machine translation support would enable users to generate a machine-translated version for all or selected parts of the source text, in order to obtain a first-pass rendering of the target translation.
- Terminology/glossary translation would provide support for translating specialized terms from translation memory.
- Dictionary widget would provide definitions for terms in both the source and target languages.

Other desirable core features included:

- Pervasive Unicode support for all input and output text, with rich conversion support in both directions. Unicode is a “superset” character encoding, with the ability to store any language or character set. Many existing tools are not Unicode-aware, creating limitations and interoperability problems.
- Ability to view alternate source text, in situations where the source has already been translated to another target language. In these situations, the tool would enable translators to view and utilize prior translations as secondary “source” for clarifying meaning and keeping translations consistent.

Workflow Features

The following features would address support for the actual processes, or workflow, of text translation.

- Progress and state management: The core workflow features would enable definition of milestones, assignment of tasks, and entry of time estimates for pending work. For both individual documents and collections of documents, the tool would provide the ability to track translation, editing, and proofreading status. The tool would also support progress estimation in both objective terms ("document translation is 80% complete") and subjective ones ("this is high quality translation").
- Role-based user features: The ideal tool would expose different feature sets for different types of users in the translation process:
 - Project managers would have a dashboard of all translation activity and status, with the ability to "drill down" for additional detail.
 - Translators would view their pending translation documents and tasks, in concert with tools to progress on those tasks.
 - Editors would view the queue of documents and document segments awaiting review, as well as the status of documents in editorial process.
 - Proofreaders and reviewers would view the queue of documents and document segments awaiting proofreading, as well as the status of documents in proofreading process.
 - Original authors would be able to track the translation status of documents they had created and made available for translation.
 - End users would be able to track the availability of translations they had requested.
- Status change notification: The platform would enable all stakeholders to be notified of changes in status to any document in the system, as well as the arrival of new documents into the system. Notification could be done via email or RSS (Rich Site Syndication).
- Accounting: The tool would be able to track hours and completed tasks for each project member, allowing managers to both assess productivity and track compensation.
- Collaborative document mark-up: Users could make annotations – e.g., "I had a problem with this phrase" – at any level of detail or scope, and invite others to give feedback. Such markup could also be tied to shared online discussions such as chat rooms or instant messaging.
- Review process: As each translation was ready for review, the tool would support assignment of review tasks, and track both editorial and proofreading reviews. An additional component would provide support for peer review, where fellow translators could assess the work and comment on semantics, nuance, and other subtleties.
- Reputation management: Hand-in-hand with a review process would be reputation tracking for each user of the system, especially translators. Such a subsystem would track the quality of each user's work, in both objective terms (100% of assigned tasks completed) and subjective terms (editors, proofreaders and peers could evaluate translators on various criteria). Such a system would ideally enable translation managers to select the most suitable translators and other personnel for specific translation tasks.
- Import and export of source documents: The tool would be able to handle the broadest range of document formats and encodings, allowing easy import of source texts from Open Office and other suites, HTML, PDF, raw text and other editing tools. Translated texts could be exported in all of the same formats.
- Segmentation of larger texts: Large documents often need to be broken down into smaller units in order to be delegated to different translators or parceled out in manageable units. Segmentation support would allow breaking large documents into such units, provide tracking of each segment's status and task ownership, and enable eventual re-assembly of the translated segments into a final unified document. Additional functionality would allow prioritizing the segments, so that important sections were done first, and less important sections could be deferred and potentially delegated to less experienced translators.
- Version tracking: Translated documents go through a number of versions, both in translation as well as during subsequent editing and proofreading. The tool would archive all versions of each document using a subsystem such as Subversion, and then provide the ability to compare any two versions to see differences and changes.
- Cross-lingual change tracking: While version tracking would maintain history for individual documents, cross-lingual change tracking would enable project managers and translators to be notified when a source document was changed, in order that other dependent language versions of the document could be flagged for pending updates. Such a feature would enable multi-language sets for a particular document to remain synchronized.

Additional Features

- License tracking: An ideal tool would be able to track licensing for imported documents, and ensure that appropriate licensing was assigned to any translated works in a system that supported human overrides to reflect the broad range of intellectual property agreements under which translations can happen.
- Offline use: While internet-based features would be critical to the realization of any “dream tool”, just as essential would be the ability to enjoy rich offline functionality. The tool would need to launch and operate when no connection was available, supporting translation and editorial tasks, and storing edits and progress updates for synchronization the next time the user connected.
- Unified translation memory: This feature would provide local translation memory combined with access to external translation memories. There are a range of memories available, but it would be useful to have centralized repository capabilities. Similar functionality could be provided for glossaries.
- Multi-lingual comparison: For documents translated into multiple target languages, this would allow translators to review how translation was done for related languages. For example, when translating to Serbo-Croatian, a translator could be aware of other Baltic language translations, and could see the work other translators had done in those similar languages.
- Pledge bank: Funding the translation of open content is often problematic, because it is not usually institutionally driven. Pledge bank functionality would allow translators to post estimated costs for translating particular documents, and allow parties interested in seeing the document translated to pledge monies they would contribute if the document was actually translated. The document would only be translated and pledges collected once the pledge total reached the projected translation cost.
- Translation of SVG graphics: Scalable Vector Graphics (SVG) are images where the data stored includes any text contained in the graphic. A dream translation tool would support translation of the text within SVG files, in order to offer a more complete translation solution.

BASIC CONCEPTS

6. Translation
7. Localisation
8. INTERPRETING
9. Dictionaries and Glossaries
10. Translation Memory
11. Machine Translation
12. Standards
13. Basic Technical Concepts

6. TRANSLATION

When we use the term "translation" in everyday speech, we typically mean that content in one language is rendered in another so that speakers of the second language can understand it. This usage is quite broad and covers everything from real-time interpretation of the spoken word to translation of a novel for publication in another language.

Translators strive for faithfulness and transparency as their ideals in translation. Faithfulness refers to how accurately the translation represents the meaning of the original text, while transparency is a measure of how readable the text is in the target language. These two aspects are often at odds with one another leading to the concept of equivalence.

Functional (or dynamic) equivalence refers to a translation that aims to represent a concept well in the target language, adapting idioms and structure to the target language. Formal (literal or direct translation) equivalence refers to a translation that faithfully follows the sentence structure and idioms. These translations would be difficult to read and idioms will lose meaning. Most often a translation is a balance of these aspects.

Indeed it very much depends on what you are translating to understand how far you can go in terms of "simply changing" things in whatever direction you wish. When translating a blog entry it is essential to transmit the same "feeling" of the original text. Dealing with different cultures one might need to go so far to completely change the text in some of its passages. When the original text instead is about facts freedom in expression becomes much less and you have to keep your text much more aligned to the original one. In extreme cases this might go so far that you may not translate a concept but leave it in its original language adding an explanatory translation in brackets, simply because what you have in front of you cannot be transmitted properly into another language without explaining it. Some contexts require to "force" the target language by trying to maintain the sentence structure as similar as possible to the original or by being limited in the length of text. The last case is mainly to be found in software strings, becoming less and less relevant for computer software but being still highly relevant for PLC messages in the machinery sector.

While equivalence is important to understand when planning for translation it is worth understanding that recent trends have questioned the idea of equivalence. Instead the idea of the purpose of a translation is now often a consideration. This is best seen in the development of a translation brief which is a document that specifies the purpose of the translation, the target audience, the reference resource, etc.

The translation brief or purpose of a translation allows translators to do things such as:

- summarisation - when the reader only wants to know what the general topic of discussion is about this is a good approach. For instance getting a summary of a Russian maths paper to see what is being considered you don't need a full translation.
- correction - correcting grammar, logic, etc.
- synopsis - a brief intro to the idea being discussed without the full coverage of a summary,

Of course many of the above concepts can introduce translator bias. Of course all translations can introduce bias from the translator, even though they strive to remain neutral. Understanding your bias is important when trying to prevent your bias from entering a work.

One aspect used to check translation quality is **back-translation**. By using this technique a translator will translate the target text back into the source language so that a client can review that the sense of the translation has remained mostly unchanged. This is used extensively in medical translation and like for the medical domain it is a useful strategy to apply in many other domains in terms of quality assurance and monitoring bias.

The traditional view is that translators should translate into their native language. The logic is that this is the language in which you have best access to idioms and expressions that you can transfer the source text into. When the language is foreign, even when well understood, it is likely that you will not have the same depth of understanding.

Translators play a very important role in society. They act as a bridge between cultures and languages. They bring new words and new concepts into their languages and culture. It is said that the close alignment and ease of transference between European languages has been achieved through a strong history of translation that has allowed many words and concepts to be shared over the ages.

Code mixing, that is when a person uses many concepts or words from various languages, is one of the aspects that is evident in spoken language but also can influence translations. Therefore in such cases, during the translation process it is important to stay aligned to the target language.

7. LOCALISATION

The *Localisation Industry Standards Association* (LISA) defines localisation as "the process of modifying products or services to account for differences in distinct markets". Thus it would entail adapting, translating and customising a product for a specific market. This would involve dealing with a specific locale or cultural conventions. By locale, we mean conventions such as sort order, keyboard layout, date, time, number and currency format.

THE DIFFERENCE BETWEEN TRANSLATIONS AND LOCALISATION

Localisation might seem identical or similar to translation. However, the process of localisation is much broader than simply translation. Localisation should ensure that the product provides the local user with the correct local "look-and-feel" while they interact with the product.

ASPECTS OF LOCALISATION

Here is a list of some of items that are considering in localisation:

1. Translation of the product's interface and documentation
2. Colours, images, graphics and icons: adapting to cultural and legal requirements
3. Rendering (can we display the text correctly, does the new text fit inside the allocated space), fonts (do we have fonts and characters for the language), bi-directional text needed in Arabic and other languages.
4. Locale data: how to display dates, time, number, currency and other regional data.

WHY IS LOCALISATION IMPORTANT?

The lack of content in locally understandable languages is one reason for the slow adoption of Information and Communication Technology (ICT) in developing countries and in today's world, access to ICT plays a major role in the overall development of a country.

INTERNATIONALISATION

When software has been properly internationalised it is ready for **localisation**. Internationalisation then is the process of adapting the software so that it can be localised.

Internationalisation would involve adaptation of the software so that interfaces can be translated, that the software makes no assumptions about the presentation of dates, times or calendars but rather present them based on the applicable locale information.

An internationalised application may be localised in those languages and for those locales that are supported by the technology used. Underlying technology, like UTF-8, defines to what extent a localisation is possible.

INTERPRETING

While translation relates to written rendering of source texts, interpreting refers to the spoken transfer of speeches or negotiations. Interpreting occurs more or less in real-time or with a very short time lack in presence of all parties involved.

Interpreters need very specific skills and a special training. They are highly qualified as they have to process two languages in "real time" at the same time as they have to transfer ideas and thought patterns from one language and culture into another.

There are a few main types of interpreting that require different specific skills:

1. **Simultaneous Interpreting:** The interpreter listens to the speaker, renders the spoken words into the target language, checks him- or herself and listens to the speaker again. This involves the ability for multitasking in the real sense of the word. This task also requires the ability to highly concentrate and can only be done for a very restricted period of time. Therefore, simultaneous interpreting is mostly done in teams. The size of the teams, among other things, depend on the languages to be covered. The interpreters sit in booths which can be located directly in the conference room or outside. If the latter is the case, the usually have sight to a video wall, so that they can see the speakers. This helps to improve the quality of the service, as they see the body language of the speaker.
2. **Consecutive Interpreting:** This task involves a special note-taking technique. The interpreter provides the audience with the interpretation after the speaker has finished with his or her contribution.
3. **Whisper Interpreting:** This kind of interpreting is used when a small delegations are involved. This is kind of simultaneous interpreting without any equipment. The skills required are the same as in item 1.

INTERPRETING SIGN LANGUAGES

Sign languages are a group of languages in the same way that spoken languages are a group. Each language is a distinct language with its own grammar and vocabulary, there are over a hundred sign languages and typically only half of the people speaking a sign language are deaf.

Many sign languages received an official status in the last forty to fifty years and from that time many have been adopted in education and in some countries news programs have added an interpreting service in sign language.

As the basis of sign languages is based in movements and not in sound, the familiar writing systems are not applicable to them. In language research there have been several attempts to come up with a way to annotate sign languages and these did not lead to a writing system that was useful in day to day writing. The SignWriting script, developed by Mrs Valerie Sutton, however has been developed in over thirty years into a script that can be used for writing any sign language.

SignWriting is being adopted in education and, research has shown that the general rule that kids who learn to read and write in their mother tongue benefit for the rest of their academic career equally applies to sign languages. This adoption of sign languages is taking place but the one big hurdle is that there are so many sign languages and they all have no material to start with. The biggest technical hurdle is that while SignWriting is recognised as a script, there is no Unicode support for it.

At this stage most of the translations done into or from sign languages is one of interpreting or the real time translation of a conversation. There are people who have started to translate the bible into ASL or the American sign language and this effectively is one of the best signs of sign languages as a language that is being written.

9. DICTIONARIES AND GLOSSARIES

Most domains have their own terminology; in this way a jaguar can be several types of aeroplane, an animal, a protein in the fruit fly or a car brand. Many texts including this manual have a glossary defining terminology used. Dictionaries and particularly translation dictionaries often do not include the specialized terminology needed in specific texts.

When a set of connected documents is translated, it is important to standardize the use of the underlying terminology, because this will improve comprehension of the translated texts. Another achievement of standardized terminology is that it can help identify hyperlinks within web based content. However, before using a term, the terminology has to be cross-checked by means of reliable resources in order to ensure usage of the correct term for the respective domain.

GLOSSARIES

Many publications contain a glossary or wordlist that includes the definition of the term as used in a specific publication. By means of these definitions it is easier to find the equivalent concepts in the target language. This helps to improve consistency in the translated text.

ONLINE DICTIONARIES

- Dict.org - Online dictionary including many resources: Webster, computer terms, etc. <http://www.dict.org/>
- Foldoc - Free On-line Dictionary of Computing <http://www.foldoc.org/>
- Wiktionary - Wikipedia like dictionary <http://wiktionary.org/>
- OmegaWiki - multi lingual dictionary <http://omegawiki.org>

10. TRANSLATION MEMORY

When you translate using a CAT-Tool (Computer Assisted Translation Tool) a database of bilingual segments is being stored. This database is called Translation Memory, abbreviated TM. Working with translation memories has two major advantages: if you are working with highly repetitive texts or on updates of translations, all the material has already been previously translated will be found in that memory. For less repetitive texts, the TM can be used to look up terminology. The advantage here is that you do not only see the term, but also how it was used in context.

When a text is broken down into so-called segments and these are translated, you have a translation memory of that text. The segmentation follows certain rules and these rules differ from language to language, because not each fullstop is actually the end of a sentence. This means segmenting rules are essential to the creation of a good and reusable translation memory.

When accepting a new job, translators are often provided with an existing translation memory of texts of the same company or of texts dealing with the same domain. In this way, the translator uses the TM to do terminology research, and when the TM is from the same company it helps to maintain the style of translation. For maintaining the style of a text, in addition to the translation memory normally a styleguide is provided.

There are both proprietary and open formats for translation memories. One of the best known standards among translators is TMX (Translation Memory eXchange), an XML based format.

LOCAL VERSUS GLOBAL TRANSLATION MEMORIES

Translation memories can be used in a range of contexts, from a personal productivity tool, to a global memory that is shared across many projects or companies. Translation memory started out as part of a desktop productivity tool, and was primarily used by individual translators to archive and re-use their own work. As Internet connectivity has become ubiquitous, translation memories are now often networked, so that many translators within a team can share their work, and more recently, with global translation memories that act as SaaS (software as a service) tools.

Local translation memories and small networks work best for teams of translators who are working for a specific client, work in a specific domain (e.g. automotive parts documentation), etc. You decide which translation memory you want to use or join based on the project you are working on, and the types of translations you're likely to need or re-use.

Global translation memories, such as the Worldwide Lexicon, collect translations from a wide range of projects and publications, spanning many language pairs and domains. This type of translation memory is not suitable for domain specific translation, but it does work well for more general content, such as newspaper articles, because the vocabulary and writing level targets a general audience.

It is also possible to combine both types, by searching first for translations from a domain specific translation memory, and then fallback to a general purpose translation memory.

EXACT VERSUS FUZZY MEMORY

Translation memory tools offer two types of searches: exact and fuzzy matches. In an exact match, the translation memory only returns translations that precisely match the source text. In a fuzzy match, the translation memory returns approximate matches. Results from a fuzzy match cannot be used as-is, but must be reviewed and edited by a translator, as even a single word can change the meaning of a whole sentence. Fuzzy matches are very useful, however, because there is a lot of repetition, especially in domain-specific material such as manuals and documentation.

11. MACHINE TRANSLATION

Machine translation (MT) is the use of computers to translate from one human language to another automatically.

At the basic level MT performs simple substitution of words in one natural language for words in another. More advanced and useful MT adapts the translation to take into account the different grammars, idioms and other language artefacts between the two languages.

In the field of translation MT can be used either to pre-translate a document in which case the translator performs a editing function to correct the suggestions from the machine. Machine translation can also be accessed through Translation Memory, in this case the translator is not correcting MT translations but is able to use the MT suggestion if the translation is appropriate.

Both Apertium and Moses are examples of a open-source machine translation tools.

There are also web services that provide Machine Translation. Google Translate is an example of an online Machine Translation service. Google Translate is not an open-source tool.

12. STANDARDS

A standard is defined by an authority or by general consent as a general rule or representation for a given entity.

Standards impact translators in a number of places. The first set allow for resources to be shared between applications. The second group define important aspects related to language that allow computers to store, display and work with languages, locales and scripts.

Standards can also include aspects related to the language itself such as spelling rules, orthography (the accepted writing system) and terminology. These could be conventions in which case they are the accepted way of doing something, or they could be official standards or resources ratified by a national language board or similar body.

APPLICABLE STANDARDS

These are standards that are of interest to translators.

Standards for linguistic variants:

- Terminology (standardised terminology for a language or domain, compare *computadora* vs. *ordenador* in Spanish)
- Orthography (different spelling conventions etc. e.g. *-ise* vs. *-ize* in English)
- Writing systems (languages may have one or more writing system in wide use, e.g. Latin and Arabic for Azerbaijani)

Standards for translation resources:

- Sharing translatable work: XLIFF (XML Localisation Interchange File Format)
- Sharing translation memory: TMX (Translation Memory eXchange)
- Sharing terminology: TBX (TermBase eXchange), OLIF (Open Lexicon Interchange Format)
- Sharing segmentation rules: SRX (Segmentation Rules eXchange)

Standards for language information:

- Unique language codes: ISO-639
- Writing systems (e.g. Latin, Cyrillic, etc): ISO-15924
- Characters: Unicode, UTF-8
- Document and content tags for languages, dialects and scripts: RFC 4646bis
- Locale information: sorting, dates, times. etc: CLDR

ADVANTAGES OF STANDARDS

Standards allow translators to use different tools and still share resources. A standard such as XLIFF would allow a translator to translate offline in an XLIFF editor, while the reviewer could be using an online tool that itself understands XLIFF. Once complete the completed translations could be stored in a TMX file and reused by any tool that can read the format.

The language related standards ensure that content creation and rendering tools are all able to clearly understand the conventions that are required by the language and ensure that the text is correctly displayed on all platforms.

ISSUES WITH STANDARDS

Standards need to be widely deployed and used to gain value, if they are not then they are of limited value. Standards are sometimes not implemented because there is no compelling reason for the standard, it is not easy to access, costly or difficult to implement.

Some parts of certainly commercial localisation are dictated by *de facto* standards such as the use of the Trados tool and insistence on the TTX format.

13. BASIC TECHNICAL CONCEPTS

Here's a quick introduction to some technical terms you are likely to encounter, as a translator of digital texts. Full definitions and explanations can be found in the section "Technical Concepts" later in this book.

Fonts

Characters on a computer screen are rendered using fonts, files that contain definitions for each character. A given font may have definitions of characters in several different alphabets or writing systems. If a font does not support a given character, text requiring those characters may be displayed as question marks or boxes on the reader's computer screen or may be substituted with a character from another font.

Unicode

Unicode is the database of characters. Older methods of managing characters restricted the computer to the use of limited sets of characters. So for example displaying Russian text and Arabic text at the same time was not feasible. The Unicode standard is intended to enable computers to display and print any combination of scripts together.

UTF-8, UTF-16

These are ways of managing characters within Unicode.

ISO 639 code

The International Standards Organization has assigned short codes to represent languages, consisting of two to four Roman letters; for example, German is represented by de and Japanese is represented by ja. You may encounter these codes when looking at web sites with translated content, particularly in the page name or the URL.

Locales

Every computer user reads and writes text and runs programs in a certain locale, depending on the default language and the geographic region of the user. The locale includes: how dates and times are displayed, default currency, how numbers are represented, the keyboard layout and other features.

RTL

Right-to-Left - the direction of the flow of text on a page. Arabic script is RTL. English is LTR (Left-to-Right).

Bi-Directional Text

The placement of both RTL and LTR on the same page. This has complex issues for many kinds of software.

Input methods

Writing in some languages is easier on a standard computer keyboard than in others. Ideograph-based writing systems, for example, require some other method of getting the text in than assigning characters to keys; these methods are called "input methods". They vary from typing a representation of the text in Roman characters, to assigning calligraphic strokes to specific keys.

Keyboard layouts

In order to type text in any given language, the numbers produced when keys are pressed must be mapped to characters in the language's script, including accents, ligatures and other markings. This is done by use of a keyboard layout. In cases where such mappings are infeasible, special input methods can be used (see the definition of that term).

TRANSLATION PROCESSES

14. Translation Processes

15. Workflow

16. The Translation Industry

17. Community Translation

14. TRANSLATION PROCESSES

It is always worth realising that the basic translation process is as simple as a piece of paper containing your source text, a pencil and a piece of paper on which to write your translation. You read the source text and write your translation on the blank piece of paper. Everything else that we discuss about the process and tools centres around enhancing this simple act of translation. More advanced processes enhance that simple translation process to increase speed, quality, collaboration and resource sharing.

The translation process can be improved with simple tools. The first resources are dictionaries, whether this being hardcopy dictionaries, electronic or online glossaries, that then allow translators to ensure that they can obtain correct word equivalents. Moreover, monolingual dictionaries provide the corresponding definitions. Previous translations of the work can be as simple as a collection of alternative translations produced by other translators. Both dictionaries and previous translations can easily be stored as books on a shelf.

AIDING THE TRANSLATOR

The first technical enhancement to aid the translator are tools that automate the roles of paper-based dictionaries and previous translations. We can use terminology lists and electronic dictionaries as equivalents to paper-based dictionaries. Previous translations collected in a translation memory are now databases of all previous translations that the translators and their respective teams have carried out. These tools can all be used to aid our paper based translation process.

Translators can enhance their paper-based mode of translation by using a word processor. Now they have access to tools like spell checkers and grammar checkers.

Taking the word processing idea one step further results in a computer aided (or assisted) translation (CAT) tool which creates an environment which appears similar to the word processing environment with integrated electronic terminology lists cooperating with a translation memory. This eliminates the effort of running different applications and results in a workbench like environment optimized for translation.

EXTENDING THE WORKFLOW OF THE TRANSLATIONS PROCESS

Translations are often performed by a team. This implies that various tasks including terminology research, pre-translation, translation, review and proof reading are carried out by different team members. When applying our simple paper and pencil process, this simply means handling stacks of paper from one person to the next as we move through the various stages of the translation process.

The workflow is one area in which extending of electronic tools has clear benefits. This can be as simple as using email to move the document from stage to stage in the process. It is quickly evident that email is not ideal when automating this process. This requires a large amount of communication overhead, different document versions can be mixed up and it is difficult to track the current stage of the process. However, email remains a perfectly valid tool to enhance the process of moving translation work from stage to stage.

However, there are more sophisticated tools available to manage the workflow of a project within a team from stage to stage. This can be a translation workbench, a globalisation server, a translation management server (TMS) or a specific project management tool for translation projects. These tools are purpose built for the translation workflow and include features that you would find in general project management and workflow tools.

A TMS ensures that the correct work is done by the correct person by means of the corresponding resources. As an example, if a team was translating 3 medical brochures then a TMS would assist in the following ways :

- The project manager would define the project, upload the three documents, define the required stages of the process, assign the different tasks to the corresponding persons, and last but not least define resources such as terminology and translation memory.
- The TMS then manages the flow of data, ensuring that the respective translators only uses the correct resources.
- When all tasks are finished it manages the handover of work from user to user.

During all these tasks the TMS's role is to optimise the flow of data to ensure that work is completed quickly without any errors induced by mistakes in the process.

BRINGING A COMMUNITY INTO THE LOCALISATION PROCESS

The translation process can be further extended by allowing communities to participate in the process. This could be to perform the full translation process or to add value to the translation process by assigning texts, reviewing texts or performing translations into languages not included in the core set of languages.

Eventually, community involvement is an extension of the TMS workflow. It involves other issues brought about with the sheer number of people involved, the number of tasks being carried out and the volume of resources. It introduces new concepts derived from crowd sourcing and social network fields such as reputation scoring and community building.

Referring to web-based tools for community translation, they are to be seen just as extensions of the CAT tool used to enhance the translators access to resources.

15. WORKFLOW

If translation is the task of converting a source document from one language to another then its helpful to understand that workflow is simply the flow of the translation task from one role player to another. It is important not to lose sight of the fact that we are still translating a document and that the workflow is just a number of steps that we follow to ensure that the task is performed with excellence.

The areas, steps or stages of a translation workflow are similar whether we are talking about the translation industry, traditional open translation, localisation or crowd-sourced translation. Whether the processes scale has more to do with the technology used and how each of the areas are implemented in practice.

TEP: TRANSLATION, EDITING, PROOFREADING

Translating the content is king and thus Translation, Editing and Proofreading (TEP) is central to the translation workflow. The translation industry developed the concept with the idea that every translation is worked on and looked at by three different sets of eyes. In reality smaller translation teams might not have such a luxury of resources.

As the number of languages grows it is easy to see how there would be an increasing communication burden and why Translation Management Systems (TMS) are used to ease and facilitate communication between role players.

In the traditional TEP process a translator will receive the work to be translated, instructions and resources. The task is that they translate this into the target language. Once complete the work is sent to the editor who will review the work. This would include tasks such as checking terminology use, language use, grammar, etc. Lastly the work is sent for proofreading where the body of work is seen as a whole and approved by the proofreader.

Major issues might result in the work moving back a stage for rework in which the tasks would then be performed again.

Any of these role players might communicate with the customer either directly or through the project manager to ask for clarity on terminology or the content.

PROJECT MANAGEMENT

If we view translation projects as a task that needs to be project managed then we can understand the supporting stages surrounding the TEP process. These stages are usually performed by different people but follow a generally linear sequence.

Contracting or Selection

How do we know what work to translate? In the translation industry that is simple, companies are contracted to perform a translation task. In the open translation world it is a little different but there will still almost always be some sort of agreement on what work is to be undertaken.

Selection of work can happen in these ways:

- The paying client selects the work
- The project needing translation highlights which areas need translation, probably prioritising these.
- The translator themselves selects whatever they feel like translating

Regardless of how the selection takes place at the end of this stage we have two things. Firstly, a piece of content to translate and secondly some sort of agreement to perform the translation.

Preparation

At this stage we are focussed on ensuring that everything is ready for the TEP workflow to begin and to run smoothly.

Here are some of the tasks that might be performed at this stage:

- Pretranslation - checking and adjusting texts so that it is easier to translate. This could include breaking complex sentences apart, correcting logic in arguments, ensuring consistent terminology use, etc.
- Engineering - it may be necessary to extract the text from some system in which case localisation engineers will transform the text from one format to another
- Resource preparation - this could involve a number of tasks. Extracting terminology from the text and preparing new terminology lists or selecting which existing terminology lists to use for the project. Selecting which translation memory resources to use. Developing or selecting style guides.
- Translation brief - writing the instructions for the translators to follow when translating.
- Assignment - delegating work and roles to different translators in different languages

The aim is that at the end of this process you have the right translators selected, that they are translating good texts, that they have the right resource to assist them and they are following clear instructions. With this in place it will help keep all languages consistent throughout the linguistic process.

Post-processing

Once the translation is complete and has moved through all of the TEP cycles there may be this last stage. Some of the aspects covered here could include:

- Engineering - transforming translations back into the original format. Of course only relevant if there was some sort of engineering at the start
- Layout - there may be some DTP work to layout the translated text.
- In product review - checking everything in the final state to ensure that all layout, engineering and other tasks have not broken anything.
- Quality assurance - usually this is part of the TEP process but it is possible to run some QA processes at this late stage to catch errors.

Delivery and invoicing

With the work complete it is delivered to the client. In the traditional model that is the delivery of the work together with the invoice. In the open content model that could simply mean that the translation is published for public consumption.

NEW METHODS AND ADAPTATIONS OF THE WORKFLOW

It is important to view new technologies that are assisting with the translation workflow in context of the previous general workflow.

Thus, crowd sourcing professional or volunteer translations is simply a different way of selecting the people to perform the roles in the various stages of the workflow. Similarly, machine translation technologies are simply tools that are selected as resources in the preparation stage to assist the translators in the TEP stage.

These workflows may need to be adapted to the needs and resources of the project. For instance if you have one Vietnamese translator then following a full TEP process is impossible, there should be flexibility with awareness of the risks involved. A simplified process can also assist other languages where they would be able to translate more work when resources are freed from all tasks of the TEP.

New approaches are also looking at involving more people in each of the stages of TEP. Thus instead of one translator there might be five. Various approaches can be adopted to reduce the risks involved. This could include allowing one lead translator and four people making suggestions. Or, five translators and one editor.

16. THE TRANSLATION INDUSTRY

Open Translation approaches stand in stark contrast to the ways in which translation has been done traditionally by professional translators.

The Translation/Localisation industry is a 18-20 billion dollar (US) industry and growing. Even in today's economy, it is predicted that the language industry will soon grow to between 30 and 40 billion dollars. As the world becomes more and more globalized, the need for communication between cultures and the need for businesses to find new markets is increasing. Surprisingly, less than 1% of all the content produced in the world is actually professionally translated.

The industry is extremely fragmented. There are 3 or 4 large companies with revenues in the \$200 to \$500 million per year range, 100 or so midsized agencies with revenues in the \$10 to \$200 million range, and literally hundreds of small "mom and pop" shops and individual translators who incorporate. Many of the smaller companies may translate only in one language pair such as English-French/French-English, and many of these can specialize in one specific field, such as medical or legal content.

This brings into play situations where the largest companies may be competing for the same project with an individual translator, so pricing can vary widely.

The language industry is most definitely a viable growing industry. While translators themselves might be reluctant to change the way they work and adopt new technologies, agencies for the most part embrace many of the technologies that keep improving such as translation memories, machine translations, and content management systems.

WHO DOES THE TRANSLATION?

It is important to note that most translation companies do not do translation. The translation company is a sales and project management organization. Large buyers of translation like Microsoft or Adobe do not want to contract with thousands of contractors; they would rather go to a large player who can manage the process. An agency does not have a Zulu or Bengali translator sitting in a room waiting for a translation to come in. Translators are hired on a freelance, project by project basis.

Translators are usually trained in a language field, certified, and they generally translate into their native language, as it is very difficult to write in a language that is not your native language. That said, there is no such thing as a "perfect translation". Translation is an art form.

Management of translation projects is very people intensive. In addition to the translators, there are many other roles. These include a large component of localization engineers who transform texts as well as graphic designers and desktop publishing (DTP) specialists who perform image manipulation and page layout tasks. The overhead for a translation agency may thus be quite high.

ORGANIZATIONS NEEDING TRANSLATION FIT A PROFILE

There are two basic kinds of translation clients.

1. **Inexperienced** - Those who have never had anything translated. Often they do not understand what's involved, what the costs are and, most importantly, that even with the best technology, it is still humans that ensure that the translations are accurate. This means that if a client has a 500 page manual and expects it to be translated, formatted and printed from English into Japanese in 24 hours, they need to be educated regarding the process.
2. **Burnt** - clients who've had a bad translation experience in the past and realize the value of high-quality translation.

The new clients are often companies that have never crossed a language barrier before, hoping to increase their business abroad. For these clients every agency needs to explain how important high quality translations are. A bad translation of a website will have people clicking out in seconds. Similarly, a poorly translated product manual can create a negative perception of the actual product.

THE NEED FOR QUALITY DRIVES THE MARKET

The main concern from the translation buyer's side is quality. Translation agencies spend money and time developing their workflow to be as efficient as possible while still ensuring the highest quality. Technology tools are needed and used, such as content management systems (CMS), workflow tools, translation memory (TM) tools, machine translation (MT) tools etc., but it is still human beings that do the quality assurance (QA). This keeps the personnel costs for the agencies high, and thus, the price for getting something professionally translated can be too high for many potential buyers. An interesting ad hoc metric used in the industry is that the cost for translating or localizing a web site is generally is about 15% of the total cost to build the web site.

17. COMMUNITY TRANSLATION

Community translation integrates the concepts of online communities and social media models to establish collaborative networks focused on creating, curating and sharing translations for various types of web content. Open knowledge projects like Wikipedia have proven that people will contribute to systems like this for a number of reasons, whether it be a passion to share, a need for the final product, or a desire to attain online stature within a community.

There are two fundamental ingredients in community translation projects: networks of individuals ready and willing to do the work, and platforms available to support the workflow. This chapter considers the roles and motivations of those participating in social translation projects, as well as the nature of the platforms that support such networks.

INCENTIVES FOR PARTICIPATION

There are a wide variety of incentives for people to participate in community translation networks, both for monetary and non-monetary reasons. People will often participate in content creation purely for fun and personal satisfaction.

Altruistic / Non-Monetary Incentives

Users contribute translations for a number of non-monetary reasons, among them:

- To make information and knowledge available to broader audiences
- As a hobby
- To share translations for interesting pages with friends or family
- To improve their language skills
- As homework
- For charity or voluntary organizations

Indirect Incentives and Compensation

In many cases, users are not motivated entirely by altruistic reasons, but can be compensated indirectly. There are many ways for publishers and community operators to pay translators without paying them directly. Translators can:

- Serve or sell ads in part of the pages they create.
- Promote their translation services or agency via bylines for the translations they create, via leaderboards, etc.
- Earn credits against other services (e.g. a travel website might offer translators substantial discounts on travel packages)
- Receive free subscriptions, travel and other in kind payment in return for meeting quality or quantity goals

Direct Incentives and Compensation

Direct payments and incentives are another tool publishers and content creators can use to increase translation quality, and improve response times. While this can include direct payments to translators, there are a number of ways to provide compensation:

- Direct per unit or per job payments, sent via online payments services like Paypal, MoneyBookers, etc.
- Payment from friends and family for helping them with translations
- An employer pays a translator to translate part of a website so co-workers can read it
- Sharing ad revenue from the translated pages they create

EMPLOYING DIFFERENT USER SKILL SETS

While both amateur and professional translators play important roles in translation communities, there are a number of other services that non-translators can provide that are equally valuable. Truly proficient translators are scarce, and should spend their time working on challenging texts. Even monolingual users can contribute to a system like this if it is organized properly.

Monolingual Users

Translations always involve a 'source' language, from which content is being translated, and a 'target' language to which that content is translated.

People who only speak the source language can participate in translation communities in multiple ways:

- By submitting and curating source material to be translated
- By disambiguating or explaining difficult sections in source texts

People who only speak the target language can participate in translation communities in similar ways:

- By scoring translated texts, not for translation accuracy, but for mastery of the target language's grammar, style, etc, something only native speakers can do.
- Detecting and flagging obviously bad or malicious edits, spam, and other posts.
- Curating translations, to help decide what the day's most interesting translated articles are.

Students and Amateur Translators

Students and amateur translators can participate in a variety of ways, by scoring translations into their languages, and if their language skills are good enough, by editing and translating easier texts (for example, by cleaning up draft translations generated by automatic translation systems).

Machine Translation Systems

Machine translation systems are often employed in translation communities to quickly and cheaply obtain draft translations for newly published source documents. These translations are not necessarily intended to be of high quality or for dissemination. If an article or text is important, people will probably begin translating and editing it. If not, few people will be reading the translation anyway, so it will be OK to leave it as is with a machine translation.

WORKFLOW AND PROCESS IN COMMUNITY TRANSLATION

Different social translation communities utilize different tools and platforms. The primary capabilities and characteristics of distributed translation systems include:

Ad hoc workflow

People create, edit and share translations via an ad hoc workflow, versus the project-oriented workflow in conventional translation systems. Someone encounters an article they like, translates it, shares it, and then others jump in to score it, contribute more edits, and so on.

Adaptive review and reputation analysis process

Distributed translation systems can invite readers to score translations, from this data identify which translators are consistently good or bad, and thus decide how much review to require for each contributor. They are geared to publish first, correct later, and often combine pre-publication review with a fast recall mechanism wherein users can quickly report bad translations, spam, etc.

Discussion and community

There is no such thing as a perfect translation, so people will often disagree about the "best" way to translate a given phrase or sentence. These systems can create forums and "back channels" where users can discuss the translations, share tips, and coach each other.

Rapid response time

Distributed translation systems are usually designed to minimize turnaround time because Internet content ages quickly. A typical system may obtain a rough draft translation from a machine translation engine, and then invite users to edit or replace these rough initial translations after their publication.

Social Features and Translation

Social translation systems often emphasize the fun and social aspect of translating, as well as encouraging people to practice translating web pages as a teaching aid. This is in contrast to professional translation networks that generally require that translators be thoroughly vetted and credentialed, and are more closed to outsiders.

FEATURES OF A SOCIAL TRANSLATION SYSTEM

A number of features are required for a platform to properly support distributed translation efforts.

Intelligent Search

A well designed social translation platform will include search tools that enable users to:

- Recommend source texts to be translated
- Search for source texts that someone wants translated
- Add a translated text to the search index (the translation may have been created locally, or may be an external resource such as a blog post)
- Search source or translated texts by keyword, tag, language, category or user
- Search for other users and translators

Collaborative Editing

The distributed and collaborative nature of these systems demands that editing tools enable people to work on translations in parallel. Wikipedia and similar systems have thoroughly explored the concept of collaborative, unmoderated content creation and editing. Similar processes can be employed in translation systems so that several translators can work concurrently on pieces of a larger document and also edit each other's work.

Translation Memory

Translation platforms typically include some form of translation memory that stores translations and their revision history on a per sentence, paragraph or document level. The details of how translations are stored and indexed will vary depending on factors that include the type of content hosted on the service and the desired translation workflow.

Reputation Metrics

Reputation is an important element of community translation systems, both for quality control, but also because it is a form of currency among users. People translate for these systems for a variety of reasons, one of which is to establish a reputation as a good translator which, in turn can lead to other work opportunities.

Online Community

Social translation systems are also, as one might expect, online communities that provide many of the features available on other types of community sites. Some are fairly simple online communities with a web editing environment and a message board, while others are sophisticated social networks and online job search tools.

ACTIVE COMMUNITY-ORIENTED TRANSLATION PROJECTS

Der Mundo / Worldwide Lexicon

Der Mundo (www.dermundo.com) is a social translation hub developed by the Worldwide Lexicon project. It is an open system, and is open to any language pair. It consists of a translation search engine and a social translation portal for RSS/ATOM news feeds.

Eco-Team (Economist in Chinese)

Eco-Team (www.ecocn.org) is an ad hoc group of Chinese students and businesspeople who translate the widely read Economist newsmagazine into Chinese for online and print (PDF) distribution.

Global Voices / Lingua Project

A global, multilingual blogging community where contributors share news and commentary from their countries. Lingua (www.globalvoicesonline.org/lingua), part of Global Voices, is a translation community where volunteer translators contribute translations for the most interesting Global Voices news and commentary to and from almost two dozen languages.

Meedan

Meedan (www.meedan.net) is a web 2.0 community focused on translating news and social media (e.g. Twitter feeds) between English and major Middle Eastern languages (e.g. Arabic)

ProZ

ProZ (www.proz.com) is an online community for professional and freelance translators. While it is not a translation community itself, it is one of the oldest and largest translation communities on the Internet.

QRedit

QRedit (trans-aid.jp) is a web 2.0 community focused on translating content primarily to and from English and Japanese.

YeeYan

YeeYan (www.yeeyan.com) is a web 2.0 community focused on translating content primarily to and from English and Chinese.

TranslatorsCafé

TranslatorsCafé (translatorscafe.com) is an online community where you can reach out for translators and agencies and ask for quotes by posting jobs. It includes discussion forums as well as tips and tricks for translators.

GoTranslators

GoTranslators (gotranslators.com) is a directory where to find professionals by language and specialisation.

CloudCrowd

CloudCrowd (apps.facebook.com/cloudcrowd/) is an online crowd sourcing application on Facebook, where translators work on small translation and editing tasks, and get paid for each task.

COMMUNITY

18. Roles

19. Community Management

18. ROLES

There are many ways you can contribute to the Open Translation Movement. You may think, for example, that if you know only one language that you cannot contribute to the translation process. However, in this case you could make yourself very useful proofreading translated content, or if the material is about a topic you know a lot about you could fact check the content. If you are a designer you may be able to contribute to the open translation process by localising, or translating, text within images.

So there are many roles within translation and finding a way you can best contribute is not always clear.

Here is a brief outline of some of the ways that you may be able to contribute :

TRANSLATING CONTENT

Translation of content is probably the most obvious role! To do this you would need to know both the target language (the language you are translating to) and the source language (the language you are translating from). It is generally considered better if your mother tongue (your 'first language') is the same as the target language. If you are translating from French to English, for example, the better results will generally be achieved if your mother language is English (although it is common practice in the translation industry to also work with translators whose mother language is the same as the source language).

It is also beneficial if you know something about the subject you are translating. It is very difficult, for example, to translate a book on a highly technical subject if you have to first look up that topic in Wikipedia. Although you can probably translate some material, such as trivial non-technical content within the larger work, it is possibly better to leave translation of this type to people that know more about the subject or to work closely with someone who knows the subject.

Lastly, if you are volunteering for a Open Translation project then ideally you should be passionate about translation or the subject being translated. Volunteering can be very rewarding if it brings you into contact with new people, or is working towards a good cause, or enables you to learn more etc. However sometimes it can be a long and lonely road and you will have to find the energy within yourself to continue. In either case it helps to be excited about the job at hand!

LOCALISING CONTENT

Some content may need to be localised by replacing currency, date formats etc. with the appropriate local equivalent. If the person translating the content has done a good job then it is quite likely that the localising of the content has already been done, however proofing localisation issues is always a welcome role.

TRANSLATING IMAGES

Images might need to be localised or translated. Localising an image means changing the image to suit the cultural context of the content. For example, if an image within an educational essay shows school children of mixed sexes, it may be necessary to change that image to represent a single sex school depending on the country and culture of the target audience.

Hence photographers or illustrators can contribute to Open Translation. Additionally if you are proficient with an image editing software you may be very helpful making images which have text and reworking the image with a translated text.

PROOFING

Reading through translations and correcting errors is a very well established and extremely important role in translation. Reading through content and checking accuracy, grammar, and spelling will not only improve the quality of the content but it can also be a very good moment to provide feedback to the original translators so they can improve their practice.

While a good knowledge of grammar for the target language is useful it is not always necessary. Reading content for 'readability' or accuracy within a domain is also the role of a proof reader.

DOMAIN KNOWLEDGE EXPERTISE

If you know a lot about the topic being translated you may be of great use to an Open Translation project without translating a single word. Many subjects require expert knowledge that is not always available to the translators, but this information can be provided by others with expertise. Helping to create the appropriate definition of technical terms is just one example where those with an area of expertise can assist translation.

TOOL CREATION

If you are a software developer then there is plenty for you to do! Find other developers working on Free Software tools for translation and join the movement. Additionally, you could always add to the documentation of a free software tool.

COMMUNITY MANAGEMENT

Open Translation is often community based, and communities do not just 'grow themselves' - they require a lot of careful attention, management and leadership. Finding or becoming a good community leader is rarely easy. People often 'find' themselves in this role without applying for a position or having an ambition to be a community leader. If you find yourself in this position then there is very little we can tell you about the role as each context requires a unique mixture of skills and personality, established connections, domain knowledge, language and communications skills, and luck.

PROJECT MANAGEMENT

Unlike the commercial translation industry, there are very few cases where Open Translation communities or projects work with Project Managers. However, occasionally Project Managers may get involved, in which case you may find yourself in the position of having to establish and manage toolsets and workflows, manage quality control, manage paid staff and volunteers, recruit paid and volunteer translators, preparing budgets etc.

A word to Project Managers that are looking to build a team - while you cannot control the skills of a volunteer translator, it is helpful to know what skill set a good translator might have.

- Mother tongue speaker, or similar skill level in the target language
- Passionate about translation
- Computer experience, in order to use translation tools

There are other nice qualifications to have:

- Graduate qualifications: Linguistics, language, translation studies
- Knowledgeable in the domain being translated

Mother tongue speaker - A person who speaks the target language as their mother tongue or is proficient at that level is the 'ideal' translator. It is important though to remember that there is more to translation than being able to speak the target language.

Many people say they are proficient in multiple languages. Some people are, most people aren't. The problem with people with multi-language skills is that they often do not have access to a deep understanding of either language. And a deep understanding is what you need if you want to transpose the ideas and language from the source document into the target language. So treat the polyglot with caution until proven otherwise.

Passionate - This may seem redundant but a person who is passionate about their language, the source material, and/or the target audiences is more likely to be able to sustain the effort needed for translation.

Computer Experience - Someone who has no computer experience will have difficulty using computer translation tools. You need someone who can comfortably use a computer and other online resources.

Graduate Qualifications - This is a nice skill to have as it helps to have a deeper understanding about language, the objective of translation and skills that can be used to find and develop equivalent terms.

Domain Knowledge - Someone with knowledge of the domain being translated has the advantage that they are able to understand the source terms and ideas so that they are better equipped to adapt the text and find equivalent terms.

19. COMMUNITY MANAGEMENT

Since Open Translation is an emergent field, examples of best practice community management are rare. However it is possible to learn a lot from the history of community development especially as it has emerged in the free software and free content sectors (there is much research available on these topics at the Open Source MIT Research Community - http://opensource.mit.edu/online_papers.php)

One of the keys to success lies in developing an understanding of why people would like to contribute to your project and methods to grow their involvement, and keep them satisfied and involved. In some cases these approaches might be very simple - for example, you may simply have the funds available to pay people so some or all of the contributors will be motivated through remuneration. However it is often the case that online communities consist entirely of volunteers or they simultaneously consist of full time staff, part time staff, and volunteers. For this type of community a different approach is required.

MANAGING VOLUNTEERS

Volunteers can join a project for a variety of reasons - they might think your cause is a good one and want to help, they might love the topic, they could enjoy interaction with other community members, they might wish to become better at translation, they might be looking to do something with their skills during boring office hours. Each of these factors may come in to play and have different weighting according to the individual. Hence, it is difficult to come up with general rules about how to attract and maintain volunteers. However experience in free software development has shown that there are some things that you can do that might help (Karl Fogel has an excellent chapter about this topic in his book 'Producing Open Source' : <http://producingoss.com/en/managing-volunteers.html>).

Many projects actually appoint a community manager. Actually, most volunteer communities have someone acting in this role even if they do not realise it. At the beginning of a project this role is probably fulfilled by the person or persons that founded the project but as a project gets larger new people may need to be found to manage the larger capacity of work being done by the organisation and so new community managers may emerge from the volunteer base of they may need to be appointed.

DRAWING PEOPLE IN

It might take a long time until you get your first voluntary contribution; when it comes in, it is exciting and could well be a day you remember for a long time! After that point the process of building a community can be slow. To draw people in you need to have an inspiring story and ways to get the message out. However, once people know about who you are and what you do, only a small percentage will offer to volunteer. When they do you should be ready with a clear entry path for them so they can get involved quickly and feel productive. You can then draw them in bit by bit and grow their role over time. When they become proficient then experienced volunteers can help mentor other new comers.

You should also think of every user of your service as being a potential volunteer. There are many tricks to this. For example, if someone finds a spelling mistake on your website you might write back with a humorous message saying "congratulations, you found our hidden spelling mistake - we have found this is a proven method for identifying good proof readers - would you like to help?" Personable and humorous messages like this can be very effective in getting people interested in being involved.

MAKING IT SOCIAL

In addition to communicating what the organisation does, a community manager may need to do a lot of communicating with the volunteers themselves. This might take place through email, blogs, chat applications etc. It could even involve old technologies like the telephone or a room with table and chairs.

With all communication, whether in real space or done via technology, remember that most volunteers don't have to do anything. They are present because of their own good will, and there are likely other things they could do if they weren't helping you achieve the goals of your organisation. Hence it is always good to keep this in mind and make the process of being involved a positive and engaging social experience that they will want to continue to engage in. This social experience might simply be a rewarding one-to-one communication with you about specific tasks to be done, or it could consist of more fun and frivolous chatter. Whichever the case, it is up to the community manager to work out which method works best for whom.

Over time, capacity may become an issue for the community manager. When a community grows beyond a certain point it is sometimes very difficult to stay current with all correspondence with all volunteers. In this situation it is good to draw upon the others in your core group to assist. If these core group members are also volunteers they may welcome the trust you put in them to help fulfil this role.

In communications with volunteers it is also always a good idea to provide encouraging and positive comments and to illustrate a clear understanding of the work the volunteer has been doing for your organisation. This is particularly true if the community manager is also the founder, or an esteemed member of the community, since many volunteers are sensitive to the opinions of people in this kind of position.

It is also a good idea to congratulate and praise specific volunteers for their work in public and private channels.

MANAGING DISRUPTION

It is sometimes also necessary to manage disruptive people. This should be done, where possible, on a one-to-one basis with direct and clear feedback or directives. Occasionally it might be necessary to communicate issues through community communication channels but you should be very careful doing this. Generally speaking, only light matters of protocol violation should be politely pointed out through normal channels. Truly sensitive communications should be personal wherever possible. If you feel you are about to send a turgid communication through a community email list, for example, it is often better to step back a little, give yourself time to cool down and then re-evaluate your approach.

Occasionally you may make a mistake and blame someone for something that was not their fault. In this case an apology is always necessary, especially if the original communication was open to the scrutiny of others.

CROWDING OUT

It is often said that you cannot pay volunteers for fear of demotivating them. This may seem unintuitive - why would anyone mind being paid? However in economic theory there is a phenomenon called 'crowding out' - it refers to exactly this kind of issue. One study in particular looked at a school where parents organised a volunteer schedule for picking up children after school. Parents would share the task of picking the children up after school and delivering them to their respective homes. In this case study the school thought this a very good idea and decided to pay parents to continue to do this. What they found was that parents, who were originally motivated by the idea they were doing something for the good of the community, now felt obliged to do the job and became resentful and less inclined to participate. This is known as the Crowding Out phenomenon and it is thought to also apply to online volunteer communities.

However it is also unclear how this factor is effected by other factors. For example, establishing a cultural norm early where it is ok to be occasionally paid might in turn counter the Crowding Out effect.

The point here really boils down to two simple truths. The first is that issues surrounding payment need to be thought through and it might not always be that 'norms' like 'being paid is good' have the expected results. Secondly, if you are a community manager in the field of Open Translation then you will need to form your own theories and try them out.

COMMUNICATE WHAT YOU DO

Most volunteers wish to know that the work they do actually has an effect. For this reason it is always a good idea to communicate loudly what your organisation does and how the work of the volunteers positively effects these outcomes.

QUALITY CONTROL

20. Quality Control

21. Quality Metrics

20. QUALITY CONTROL

When a text is translated, depending on the audience for the translated text and depending on the importance of maintaining the correct information, there is a need for ensuring quality or for assessing the quality of translations. When there is a need for ensuring quality, it is in the choice of tools and procedures that quality can be expected.

QUALITY ASSURANCE

There are many ways to improve the quality of translations. It is very much defined by the translation process. On the low end, machine translation will provide a fair idea of what the text is about while on the high end during the pre-translation, a glossary was created to ensure consistent the actual translation was done by a certified translator known for expertise in the domain using a CAT tool, the text is proofread by another certified translator or by a native domain expert.

Tools like online dictionaries and CAT tools even machine translations help translators. Making such tools part of a translation platform for volunteer translators does not only aid quality assurance, but the translators will likely be very grateful!

Translation itself is often only a part of making information available. In Wikipedia for example, the original text is often not much more than the inspiration for the article in another language. When there is a need for a more exact translation, it is important to stress this need and a more formal setting helps.

QUALITY ASSESSMENT

When you have a text, translated or not, you can assess its quality. There are many levels of assessing a text; a domain expert assesses information, a linguist assesses the language used, anyone can assess if a text can be understood and someone bi-lingual can assess the translation.

To assess volunteer contributions it is important to appreciate the availability of the skills involved. It is harder to find people willing to translate in Tagalog or Ossetian than people for Spanish or Russian. When people identify themselves with a language, it helps when they can self-organise themselves. When tools that support the translation process, people can choose to proof read and comment on the text and the translation as well as improve the text.

Readers of a text can be asked to comment on a text. A comment can be anything from a thumbs up/down, a five point vote to the commenting systems made popular by wikis.

21. QUALITY METRICS

Knowing the relative quality of translations is hard to assess. When texts are translated into multiple languages it is reasonable that many of the target languages are not known by the people seeking quality metrics.

There are a number of ways to do measure the quality of translators and translations. Depending on the type of publications and the user community, there are several methods that may be used.

- Editorial or top-down reputation tracking, where you have "superusers" that supervise or police system activity. This is a top down method of watching and assessing the quality of work contributed by users.
- User-provided reputation tracking, where large numbers of users vote on documents, translations and other translators. This is a form of distributed peer review that can generate a large amount of statistically useful data.
- Self assessed reputations, where people provide metrics about their abilities or about the quality of their work.

There's not really a single solution to this issue as each method has its strengths and weaknesses, a combination is usually best. The results are used to

- learn which translator(s) consistently submit good or bad work.
- generate rules for allowing or rejecting translations from specific users or user populations.
- learn where users are coming from, what languages and topics they are interested in.
- identify and deal with suspicious or malicious behaviour, edits, robotic scores, etc.

CASE STUDIES

22. Global Voices Lingua

23. Wikipedia

24. FLOSS Manuals

25. OLPC and Sugar Labs

26. Yeeyan

22. GLOBAL VOICES LINGUA

Global Voices (<http://www.globalvoicesonline.org>) is an international network of bloggers who translate, report on and defend blogs and citizen media from around the world. Since 2005 the project's contributors have posted summaries and reports of what bloggers and other producers of citizen media from around the world are discussing. As of June 2009 Global Voices has published over 50,000 post and its community comprises over 300. Global Voices' content is freely available for re-distribution and use in derivative works under a Creative Commons Attribution license. The project actively promotes the reposting of its content by both citizen media organisations and mainstream commercial media.

The seeds of Global Voices' social translation project "Lingua" (<http://globalvoicesonline.org/lingua/>) were sown when a group of Taiwanese fans of the site began independently translating articles into Chinese and posting them online. As their body of translated work grew, they decided to organize it into Chinese-language mirror of the site. Interactions between members of the Global Voices community and the Chinese translators made it clear that there was impetus and excitement around the idea of creating multiple similar translation sites for other languages. Global Voices set out to build a scalable infrastructure that would enable the organization to create sites for new translation communities as these came into being.

CHOOSING A DISTRIBUTED NETWORK OF SITES

The decision to use separate sites for each language group rather than integrated translation in the content management system (CMS) was based on several factors. For one, WordPress, the CMS used for the site, lacks content translation support in the core code and the available plugins either lacked the relevant features or did not permit the level of scalability required (at the time of writing in June 2009, Global Voices has over 20 language sites). In addition, the existing translation plugins were all pet projects of individuals, which meant there was no guarantee of long term stability (WordPress core upgrades invariably break complex plugins, so the programmers responsible need to remain vigilant over time).

It was also important that the translation project be kept as simple as possible so that it could be replicated and updated quickly. Integrating the translations into the main site with its thousands of posts, hundreds of users and years' worth of hacks and customizations would have been a slow process involving lots of testing before it could be launched. By contrast, the use of external sites to house the translations allowed the translation system to be effectively 'beta' released with no serious risk of exposure to embarrassment if issues arose on the main (English-language) site. A simplified version of the site's front-end design template was used for the translation sites, speeding up launch time and avoiding the design conversions that would have been needed in order to internationalize the theme. The translation template was also much less dependent on the content skew that was taken for granted in the design of the English site.

Despite the decentralized nature of Global Voices' translation section, the sites still needed a means of recording and displaying the source-to-translation relationships on posts and in the database. To this end, a pingback model based on blog trackbacks was used. This allowed translations to be associated with the source post, and the relationship to be recorded in the translation database in the process. This involved entering the URL of the source post in the editing interface of the translation site, which pinged the source when the translation post was saved. Metadata fields for original author name and original author profile URL were added to the translation interface so that both the translator and the original author were credited separately on the translation post. The system was improved over time with enhancements that simplified the process, such as the addition of a hidden section in the post-viewing interface of English-language posts where all post metadata was easily accessible to translators.

PROS AND CONS OF THE DISTRIBUTED SITES MODEL

Even though the use of external sites was primarily a practical decision made for technical reasons, many within the Global Voices community consider the separation to be one of the project's greatest strengths. While integrated translations are appealing in many ways, and while some people complain that the translations are "ghettoized" in relation to the main site, separating the content and translation communities by language has in many ways helped motivate translators to remain committed to the success of their translation communities. Having administrative rights to the site appears to give editors a sense of ownership over the content. In the case of new or less active translation sites, the separation has allowed the community to grow organically and at its own pace, rather than being shoved into the noisy and crowded space of the English-language site.

Despite the Lingua project's apparent success at facilitating both participation by translators and the act of translation itself, the system was lacking in a number of ways on both the conceptual and technical fronts. For one, the system was entirely centralized on the English-language site and its database. Translations could flow only **away** from English and not the other way around. This was partly an editorial decision, but some translators had in fact found ways around it. The English-language site was also the only one with access to the translations database, meaning that the translation sites were unable to identify versions of a post other than the English source. These technical limitations were often interpreted as a form of Anglocentrism on the part of Global Voices and considered to be misaligned with the organization's values.

REVISING THE MODEL: DECENTRALIZING POWER AND SIMPLIFYING WORKFLOW

After considerable discussion, and once it became possible to dedicate the programming resources required for the task, Global Voices' Technical Director began re-designing the system architecture to solve the problems outlined above. Though still under development at the time of writing in June 2009, the new system is designed to manage translations using the same basic model as before, with separate sites pinging each other to record translation relationships, but in far less centralized way. The database, rather than being part of the English site, was moved to a separate space and can be accessed by all of the sites in the network, which can now save and fetch translation data directly from it. The English-language site is no longer the hub for all content translation, but merely a member of the translation network alongside the various language sites. Under the new horizontal structure, any Global Voices site can be the source or destination of a translation. The new structure also facilitates translations of translations (e.g. en->fr->zh) by translators who may be interested in a story that was not written in a language they know.

Alongside these changes the Global Voices Lingua project has worked on simplifying and streamlining the translation process, notably to make better use of the pinging system that automates the migration of content and metadata from the source site to the translation. The necessity of copying and pasting content and re-selecting categories from the original post has been the most common complaint from translators, as well as the cause of inaccuracies in translated posts. Under the updated system, the source site receives a ping from the translation to which it replies with serialized data about the post that the translation site can use to auto-populate the relevant fields in the translation interface. This makes it easy for translators to create a perfect copy of the source post on their site before starting their translations.

The decentralized pinging model for translations is perhaps the one that should have been used from the beginning, but waiting to see what problems arose with the original system made it possible to identify the biggest issues between the system and our community before making large-scale modifications. It is also worth noting that in many ways the older system, as simple and insufficient as it was, may have been the right tool for getting the Global Voices Lingua translation network off the ground. Now that the project has achieved critical mass it has

become clear that all sites should be treated equally. However, the centralized nature of the original system allowed the managers of the main/English language site to maintain control over the translation sites and communities during the period where the community was learning how they might work and what kind of management they might require.

23. WIKIPEDIA

Most readers will have encountered Wikipedia articles in the course of searching for information on the Internet. Many of you will know that there are versions of Wikipedia in multiple languages. What is less well known, except to the people who are active contributors to these projects, is that there are over 250 language versions and that each language version is an independently run project with its own policies, administrators, and content.

While they all have the same aim, some are better at providing information than others. The English language Wikipedia, with around 3 million articles, gets half the traffic. The Buginese language Wikipedia, though, has only 83 articles and may require you to install a font just to see them.

Each project is addressed to the speakers and readers of its language; this means that the content is written in that language, but above and beyond that, the articles are written from the cultural perspective(s) of the peoples that speak that language. For example, the English language Wikipedia will cover topics from the various perspectives of English-speaking populations from the United Kingdom, Australia, the United States and so on, the Greek language Wikipedia will cover articles from the cultural outlook of the inhabitants of Greece and Cyprus, etc.

This does not mean that every Wikipedia community must write all of its articles from scratch. In fact there is a vast amount of content sharing between the projects which often involves the translation of articles in whole or in part. In particular, articles about historical events or living persons from a particular region are more likely to have an article chock-full of good information in the Wikipedia of the language of their region. In some cases an editor may choose to translate the entire article as is, leaving it for others afterwards to adapt it or add information that makes it relevant to local readers; in other cases the editor may translate only parts of it or may write a summary or adaptation. Once the article has been created on the local project, it has a life of its own and is treated like any other article as far as editing, expansion or even deletion.

TRANSLATION PROCESS

While all the Wikipedias are autonomous and define their own policies, the reality is that the English Wikipedia is the model that most projects emulate and the content of the English Wikipedia inspires the creation of many articles. There is a continuity in this from people literally translating an article to people reading the article and either writing a summary or re-writing the article in their own words.

There are no formal mechanisms for splitting up a text and parcelling it out to multiple translators. Since the translator community consists of all users, registered or not, who choose to translate part of an article, there is no mechanism for communication between all members of the translator community concerning a particular text except by adding content to the text itself. People who translate on a regular basis tend to set up informal communication mechanisms among themselves, usually by email, irc or skype.

The choice of what material is translated, re-written, or written is completely left to the people who volunteer to create these articles. As a result there may be a bias towards certain topics. Sadly, there are no tools that measure which missing articles have most been requested.

There is also no formal relation between articles on the same subject in the different Wikipedias. The articles are essentially different. For the readers of the article there are "interwiki links" that connect these articles. In this way a reader can learn about a subject from several cultural and linguistic points of view.

Additionally, there are technical challenges when an article is translated in its entirety. Wikipedia makes use of what is called "Wiki syntax", which allows for the creation of "templates" and "info boxes". These constructs require localisation and this requires the kind of expertise that is associated more with programmers than with translators.

THE TECHNICAL PROCESS

So, just how does this process work? What is the Wikimedia workflow, how do we coordinate translation among multiple editors, how do we deal with attribution (giving credit to the authors of the original article)?

The process will differ from one language community's Wikipedia to another; recall that each project is autonomous. But typically the procedure works like this:

- The editor decides that he or she wants to translate some or all of an article from another Wikipedia version.
- The editor makes a request for the importation of the article from the other Wikipedia. The import mechanism brings over the full text of the current version of the article and all previous versions, as long as the date, time and author of each version. In addition the discussion page of the article may be brought in; this can contain discussions about the title, removal or alteration of controversial content, pointers to external resources, and so on. It may also contain attribution information in special cases as we will see below. The request for importation may be made formally by leaving a message on a page for requests to administrators, since importation is a privileged operation not available to all users, or it may be made by leaving a note on the talk page of an administrator that the editor knows well or believes is active at that moment.
- The article is then imported by an administrator who specifies the language, the project name and the page name for import, clicks a button, and waits. In some cases, when the article for import has been edited many times, the import may fail. If repeated attempts fail, the article must be copied in by hand, and the version attribution information must be copied in by hand as well. Typically that information will be placed on the newly created local article's discussion page for reference.
- Now the work of translation can begin. There is no formal workflow management mechanism; projects have developed their own informal methods for doing this. One typical approach is that the editor can tag the newly created article with a template at the top of the page which says for example "This article is being edited for an extended period of time. Please do not edit while this notice is visible." The reader might wonder how, among the many articles on a project, someone would happen to choose to edit the very one that has just been imported, but that's part of the way that Wikipedia and the other Wikimedia projects work. A new editor may often have no idea where to jump in, given the huge pool of information available for editing or cleanup, so they may often check the page listing "Recent Changes", which shows up to 500 of the most recent edits, including article creations.
- The editor doing the translation will now proceed to edit the article online using the edit form presented to them in the web browser. (Smart editors doing long texts will edit off-line, saving their work frequently, and cutting and pasting the results into the form when they are ready to upload.) Depending on the length of the text and the editing style of the contributor, they may save after translating one or two paragraphs, after every sentence -- though this is frowned upon because it fills the recent changes list with many small edits -- or after translation of the full article.
- Notice what has not been mentioned here. There is no translation memory, there is no central glossary, there is no mechanism for handing the translation off to a proofreader, there are no quick links of terms to multilingual dictionaries, nothing. Our translators are by and large volunteers who do not have access to professional tools, so they are basically on their own. A given language project may have a glossary of terms with suggestions for translations, but there is no requirement that the glossary be followed, and indeed there is no way that such a mechanism could be enforced, unless an editor checked each translation against such a glossary manually.
- When the editor saves the article, they comment out any untranslated text remaining, using HTML-style comments (`<!--` and `-->`).
- Once the editor has saved the part of the translation they intend to complete that day, they remove any notices or templates from the page, opening it up to others for editing. Additionally they will usually add a template at the bottom of the page or on the discussion page which provides a link to the original article as the basis for the current page. Editors go through the import process in addition to providing a link back to the original article because there is no guarantee that the original article or the version of the article that was translated will not have been moved or deleted when a reader views the translation in the future.
- Another editor may notice the article in the recent changes list and may decide to work on another piece of the translation, if there are other parts to be done. There is no requirement that they do paragraphs in order or indeed that any more of the text ever gets translated. Two years might go by without further changes to the article, or editors might add content based on other sources without ever referring again to the source

article.

- When multiple editors have translated parts of the same article, differences in style or the way specialized terms have been rendered in translation may make the new article a bit hard on the eyes. As with readability issues unrelated to translation, any editor is free to clean up such inconsistencies as they see fit.
- Machine translations of articles are routinely deleted from some Wikipedia versions. While the contributor who adds a machine translation probably does not speak the target language and hence has no other way to add content in that language, most communities view the addition of machine-translated content as doing more harm than good. Since all editors work on a volunteer basis, they generally prefer to work on articles of interest to them, rather than cleaning up machine-translated articles in an area that they may not know or care about. For some editors, another disincentive for editing machine-translated content so that it can be retained is that cleaning up machine translated content may be more time-consuming than translating straight from the original text.

TRANSLATING REQUESTED CONTENT

Some materials such as Wikimedia Foundation policies, surveys, or informational notices are disseminated to all of the projects, and these require translation across all projects within a relatively short period of time. These materials are typically the subject of a request for translation, handled on the project coordination site, meta.wikimedia.org. Because there is a time constraint, translators check these pages more frequently and may respond to requests fairly rapidly if the size of the translator community in a given language is large enough to permit it. No importation of text is necessary; one creates an article of the same page name with the language code appended (typically by clicking on a link to the missing page on a list of missing translations), copies in the original text by hand if desired, and then proceeds to edit. Translating off-line in these cases is less desirable, if only because another translator may be watching the same language version of the article and decide that the first editor is not actively working on the article if no changes are recorded in the recent changes list after a half hour or so. If the second editor then proceeds to translate the very part of the text the first editor was working on off-line, someone's effort will have gone to waste.

QUALITY AND QUANTITY

Many people involved in the Wikipedia community care about the quality and quantity of the articles. This resulted among other things in a "must have list of articles". This list, while well intentioned is very much culturally biased; do people who speak Buginese care about American football and would an article in Buginese have the same relevance..

24. FLOSS MANUALS

FLOSS Manuals is a non-profit online community whose aim is to produce quality free documentation for free software. The community started as an English only project but rapidly evolved to other languages. There are currently 5 language implementations of FLOSS Manuals; English, French, Dutch, Finnish and Farsi. Of these the English site is by far the most active, and second to this would be Farsi and Finnish. As of June 2009, FLOSS Manuals has approximately 1000 registered contributors with 750 of these registered only on the English site.

Each implementation of a FLOSS Manuals language is a stand-alone installation of the FLOSS Manuals platform (which is all open source) and can be found online at the following URLS :

<http://fa.flossmanuals.net>, <http://fr.flossmanuals.net>, <http://en.flossmanuals.net>,
<http://fi.flossmanuals.net>, <http://nl.flossmanuals.net>

To enable Localization and Translation FLOSS Manuals has built its own tool set on top of the core platform (Twiki). These extensions are available from the TWiki.org repository and they enable some very useful functionality. The two plugins are called Xchange and Localize.

XCHANGE

Xchange enables the transfer of manuals and chapters between each language implementation. The basic mechanism is RSS which also opens some interesting opportunities for an RSS manual publishing mechanism which has not yet been explored.

The screenshot displays the XCHANGE web interface. At the top, the word "XCHANGE" is written in large orange letters. Below it, the interface is organized into three main sections:

- 1. Select Server:** A dropdown menu labeled "Select site".
- 2. Select Manual Source:** A dropdown menu with a "Transfer Manual >" button next to it. To the right, under "Existing Manuals", is a dropdown menu labeled "Select manual".
- 3. Select chapters (click to select):** A large empty box with a vertical scrollbar. To its right, under "Existing Chapter List", is another large empty box with a vertical scrollbar. A "Transfer Chapter >" button is positioned between these two boxes.

At the bottom of the interface is a section titled "Chapter Preview" followed by a large empty rectangular box.

LOCALIZE

Localize is a Portable Object file editor (.po file). PO files are used to translate the interface of FLOSS Manuals. When a new language community wants FLOSS Manuals the interface is translated using Localize.

TRANSLATE					
LANGUAGES					
Translation file	Translated	Untranslated	Fuzzy	% translated	
da.po	674	0	0	<div><div></div></div>	download
de.po	674	0	0	<div><div></div></div>	download
es.po	610	39	25	<div><div></div></div>	download
fr.po	594	40	40	<div><div></div></div>	download
it.po	673	0	0	<div><div></div></div>	download
nl.po	661	10	3	<div><div></div></div>	download
pt.po	674	0	0	<div><div></div></div>	download
zh-cn.po	670	0	3	<div><div></div></div>	download
zh-tw.po	593	40	41	<div><div></div></div>	download
ko.po	0	673	0	<div><div></div></div>	download
New translation					
Name:	<input type="text"/>				
	<input type="button" value="Create new translation"/>				

TRANSLATE HOLDING ZONE

FLOSS Manuals prefers to work towards establishing autonomous language communities. This is because manuals need to be kept alive and grow. If a manual is simply translated without a community to maintain it then the life span of the manual is limited. Therefore, it is preferable to establish language communities that create original material as well as translate and maintain material.

However, there are often requests to translate a manual into a language for which there is not yet an established FLOSS Manuals language community. To enable these translations FLOSS Manuals has established a 'Translation Holding Zone' - located online at <http://translate.flossmanuals.net>

Currently there are manuals in this zone that have been translated, or are in the processes of being translated, into Arabic, Romanian, Spanish, Greek, Russian, Myanmar, Afrikaans, Portuguese, Brazilian Portuguese, Polish, German, Mandarin, Hebrew, Vietnamese, Turkish, Italian, Japanese, Hindi, Catalan and others. For a complete list see : <http://translate.flossmanuals.net/write>

The theory is that once there has been significant activity in any one language in the Translation Zone, then the content and community to can establish a FLOSS Manuals language site.

WORK FLOW

When a manual is transferred between language servers it is placed in a translation work flow. This means that the manuals are editable as if they had been created with the standard FLOSS Manuals manual creation process, but there is also an additional 'translate' button.

Chapter List			Actual File Name
DIGITAL FOUNDATIONS			
INTRODUCTION			
INTRODUCTION	edit translate	complete	Introduction
INKSCAPE			
THE METAPHOR OF GRAPHICS APPLICATIONS	edit translate	complete	TheMetaphor
SYMMETRY	edit translate	complete	Symmetry
TYPE ON THE GRID	edit translate	complete	TypeOnTheGrid
COLOR THEORY	edit translate	complete	ColorTheoryAndBasicShapes

This button enables access to the translation view of any chapter - which is a page where the chapter can be edited. The edit window displays alongside the original source material. Translators can then see the original source and browse forward to any newer revisions of the material to check the translation against the latest version of the chapter.



When a translator has finished working on a chapter they can mark it with any one of a number of status markings including 'to be proofed', 'needs images', 'untranslated', 'needs updating' etc.

Cancel

Save

Unpublished ▾

QUALITY CONTROL

In theory each manual has a Maintainer and it is their job to check the quality of the content before pushing the 'publish' button. The publish process copies all dynamic content to static HTML files, and generates a PDF. These published files are what the 'reader' sees and the theory is that no content will be brought forward into reader space unless the manuals are of very good quality. However, the reality is that not every manual has a dedicated or pro-active Maintainer so this process is not well regulated in some of the sites. We have found however, that as the size of the community grows this process improves.

While this work flow is not sophisticated it at least works technically and is reasonably easy to use.

Additionally, each language community has a community manager and it is their job to recruit and build an ecosystem of contributors. This means working out who is translating, what the quality of the translations are, and how to build an environment to improve the translations.

The theory is that each language community will manage its own quality control. In the English site the quality is managed by the contributors checking each others work and each manual has a Maintainer whose job it is to keep an eye on quality issues and develop strategies as necessary. For the creation of original material on the English site this works very well however, as mentioned above, there is not yet enough participation on the other language sites to gauge the effectiveness of this strategy for translation although we have had comments that some manuals have been very well translated and very few comments to the contrary.

TECHNICAL ISSUES

The largest technical issue we have faced apart from developing the Localize and Translate plugins, has been to work through the issues for supporting bi-directional text. This issue arose during the development of the FLOSS Manuals Farsi site, and since the technical developers were not familiar with right-to-left layout, nor how it works with left-to-right text when displayed on the same page. As the technicians were working remotely with the Farsi management team this process was very confusing at first and required a lot of patience, communication and testing.



Thankfully FLOSS Manuals has resolved these issues and can now support authoring in a bi-directional environment, and the rendering of this content in HTML and book formatted PDF.

Unicode support for our PDF engine has now also been resolved.

THE FUTURE

FLOSS Manuals is developing a new platform for collaborative authoring and will investigate the integration of other Free Software tools into the translation tool chain, including Pootle and World Wide Lexicon. Work flow elements will be developed however, the issue of translator management and quality control will still largely remain in the domain of community and volunteer management.

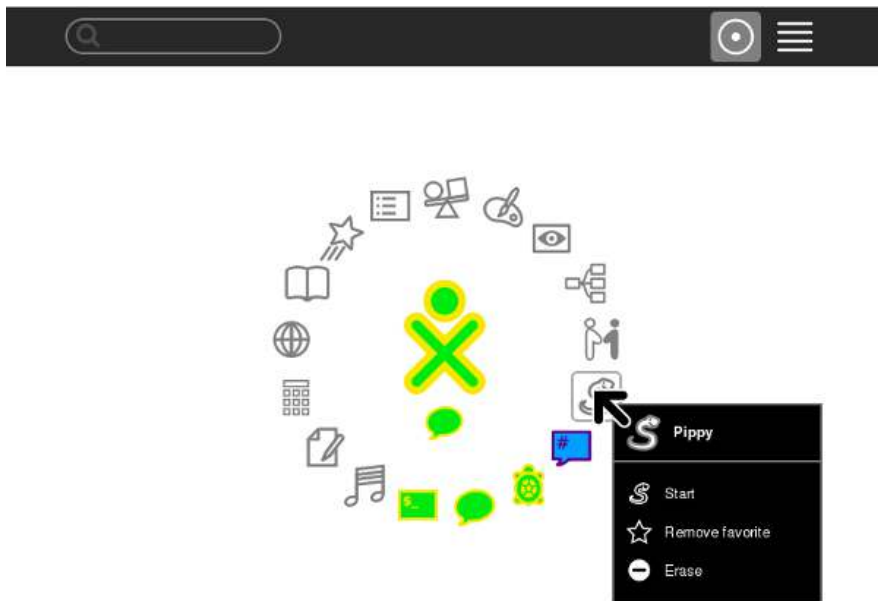
25. OLPC AND SUGAR LABS

One Laptop Per Child is an education project, initiated by Nicholas Negroponte of the MIT Media Lab. OLPC created the extremely low power, rugged XO, inexpensive computers for children around the world and began development of a suite of Free Software for education called Sugar, since taken over by Sugar Labs.

The aim of OLPC, Sugar Labs, and their many partners is to :

...create educational opportunities for the world's poorest children by providing each child with a rugged, low-cost, low-power, connected laptop with content and software designed for collaborative, joyful, self-empowered learning...

Here is a screen shot of the home view in Sugar, with a circle of selected Activity icons. You will see that there is no text visible, except in the tooltip shown here for Pippy, the Python programming editor for children. The central XO icon shows this user's colors, which also mark icons of Activities last started by this user (in this case, Terminal, Chat, and Turtle Art). The one icon (for IRC) shown in other colors was started by another user on a different computer, and shared with the community, including this user, who joined it in progress. Further explanations of icons, icon coloring, and other UI elements can be found in the FLOSS Manuals *Sugar* manual at <http://en.flossmanuals.net/sugar>.



In addition to the languages currently available, OLPC and Sugar Labs are working on localizing Sugar software to about 65 more languages, listed at <http://dev.laptop.org/translate>.

Other translation activities include

- The OLPC Wiki(<http://wiki.laptop.org/>)
- The Sugar Labs Wiki <http://wiki.sugarlabs.org/>
- Educational content from many sources
- Textbooks and other learning materials
- Teacher training materials
- Materials for governments considering putting laptops and Sugar into their schools.

LANGUAGE SUPPORT

The OLPC XO has its own hardware layout, so each of the Linux keyboard files has to be adapted to it. These modified files have been submitted to the upstream Linux distributions, including Red Hat and Debian and have made their way to general availability. The XO does not have a Caps Lock key (at the request of a child) and has an extra key for × and ÷, which are not in ASCII, but are strongly needed for elementary school arithmetic. The Wiki page section http://wiki.laptop.org/go/Keyboard_layouts#OLPC_keyboard_layouts links to pages for the language-specific layouts.

Changing the User Interface language is done through the Sugar Control Panel, accessed from the Home view by right-clicking on the XO icon, and selecting it from a menu. In the Language Control Panel there is a menu offering all of the languages installed in the Linux Operating System, whether or not Sugar supports them in Activities. The result can be an odd mixture of English and the chosen language, or of Last-Resort characters indicating that there is no font installed for the language selected.

Sugar language support depends on a set of fonts covering the writing systems used in languages of target countries, other widely used writing systems, and some symbol fonts. For current deployments, this includes some subset of the following.

- Latin alphabet with extensions for European languages and Vietnamese
- Cyrillic for Mongolian (Mongolian script to be added when a standard appears)
- Arabic with extensions for Dari and Pashto in Afghanistan, Urdu and other languages in Pakistan and India, and Hausa in Nigeria
- Nine Indic alphabets of India for 21 of its scheduled languages
- Sinhalese for Sri Lanka
- Laotian for Laos
- Thai for Thailand
- Khmer for Cambodia
- Ethiopic for Amharic in Ethiopia
- Greek, Hebrew, Chinese, Japanese, Korean
- Math
- Dingbats

The limited storage space on XOs makes it impractical to install fonts for all supported Unicode ranges. In addition, there are Unicode ranges for which no Free Fonts are available.

It is important to distinguish between the locale set for the system overall, particularly in the User Interface, and the current keyboard and fonts that a user needs in order to deal with other languages. In general, Sugar uses standard Linux methods for selecting keyboard layouts and fonts, but plans to change from the existing xkb keyboard software system to SCIM (Smart Common Input Method), which is far more flexible. The xkb method only supports entering one character per keystroke, but there are many languages whose essential alphabetic letters are only available in multiple-character sequences in Unicode. This includes multiply-accented letters that local users may be used to typing as single letters on typewriters adapted to their languages.

Unlike the situation in the United States and parts of Latin America, children in Africa and Asia commonly need to know their own local language, a standard national language, and an international language in order to function in society, and may want to learn other languages for business, research, or other purposes. In many former colonies, as in Ghana, parts of India, and other places, a foreign language such as English or French is the language of instruction in all of the schools. Former Soviet Republics have had a difficult time getting away from the use of Russian exclusively in schools, since most available textbooks are in Russian or other foreign languages, and not in local languages such as Mongolian, Uzbek, and so on.

Making Free textbooks available in local languages is necessarily a large part of the program, although nobody has taken it on so far.

ORGANIZATION

Localization and translation work for OLPC and Sugar labs is done primarily by volunteers. Some machine translation is used to create drafts of Wiki pages, and some countries employ commercial services for their own purposes.

Some in the Sugar community propose that localization and translation be assigned to the more advanced language students in the countries concerned as homework. This can include all of the categories of work described above, plus translating and writing Wikipedia pages in their own languages or any others they are learning. The same idea applies to any other software or content the community considers worthwhile. Teachers and students can examine a set of proposed translations, and construct a collaborative translation from the best work submitted. At some point, this kind of work can be extended to business plans for graduates to take to microfinance institutions.

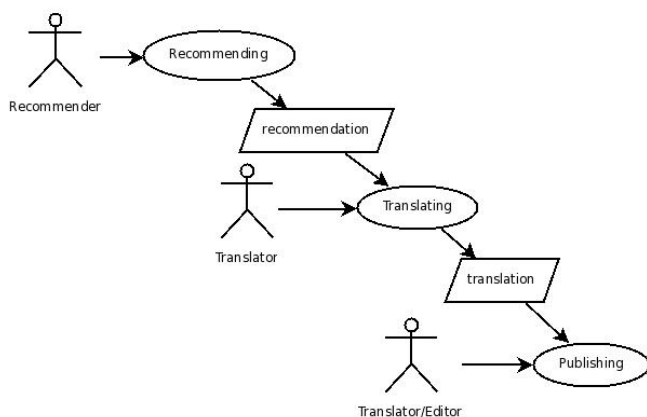
26. YEEYAN

Yeeyan (<http://www.yeeyan.com>) is the largest open translation community in China. To date (June 30, 2009), it has more than 90,000 registered users, with about 5,000 community translators who have published nearly 30,000 translations on the site.

Recently The Guardian (<http://www.guardian.co.uk>), collaborating with Yeeyan, launched their ground-breaking project - The Guardian's Chinese UGC (User Generated Content) site (<http://guardian.yeeyan.com>) on May 18, 2009.

DISCOVER, TRANSLATE AND READ THE INTERNET BEYOND YOUR LANGUAGE

"Discover, translate and read the Internet beyond your language" is Yeeyan's slogan. We are doing it by three core processes: recommending, translating, and publishing. Each process serves different needs.



The recommending process is to allow users to recommend interesting articles to the community. It is also an effective means for the site to organize the community to produce desired content. "Crowd recommending", which allows users to vote for recommendations, was one of the earliest features but removed later due to inactivity. Now Yeeyan is going to output filtered recommendations to mini-blogs, such as twitter. Together with the mini-blog API, we are expecting "crowd recommending" to become an often-used feature and help the community to identify demands.

In the translating process, the tools for translation work are actually a small part: the key task is to provide a reliable on-line rich text editor to users so that they can type their translation in the editor or simply copy/paste translations from elsewhere. The real heavy duty work of the translating process is "**quality control**". We have developed a tool to allow users to indicate translation errors by "in-line comments". Error rate is calculated based on the number and type of valid errors, which gives the quantitative metric of a translation. The formula is learned from the traditional translation industry. Other quality control tools include version control and translator evaluation. Quality control not only guarantees service quality delivered to clients, but also provides translators with helpful and meaningful rewards for enhancing their comprehension skills.

The publishing process is responsible for organizing and displaying translations, i.e., for reading purposes. A hybrid mechanism that combines a Digg-like computer algorithm and community editors' choices recommends "top" translations to readers. Selected translations are also output to mini-blogs. The publishing process will also provide service interfaces, which may include customized subscription, and most importantly, a licensing/purchasing service.

COMMUNITY-BASED ENTERPRISE

No matter how technology can help, translation itself is a labor intensive work, or in another words, humans are always at the center of a translation. This is particularly true for "open translation" when a community (or crowd) is involved.

Yeeyan is step by step building a hierarchy in the community. We have editors and proofreaders from the community to do daily recommendations and proofreads; we even have a column editor from the community to take a major responsibility for The Guardian's Chinese UGC site. Currently, above 50% of Yeeyan's gross income is rewarded to the community, even though the company is still fighting for its survival.

The current structure is still centralized, which means Yeeyan has a direct control on all commercial projects. For non-commercial projects, Yeeyan provides a wiki platform (<http://pro.yeeyan.com>) to allow users to play on their own. A very successful project organized by users is the translation of materials from the Leukemia and Lymphoma Society (<http://pro.yeeyan.com/wiki/%E8%AF%91%E7%88%B1>). The project's name means "Love by Translation". It has finally gained sponsorship from Google's charity fund.

The centralized structure is just a middle stage. By transplanting project mechanisms from inside the company into the community through product development and training the community, we are aiming at building a decentralized, self-organized community, which allows users to run commercial projects on their own.

Yeeyan has been complained for ineffective communication with the community for quite a long time. Active users submitted suggestions, opinions and bug reports to Yeeyan's BBS (<http://bbs.yeeyan.com>) but rarely received prompt feedbacks. Yeeyan's blog (<http://blog.yeeyan.com>) was not updated regularly. Even product releases were not announced and explained to the community in a timely manner. To improve this, Yeeyan has emphasized within team that product-related posts such as bug reports on BBS must be replied as soon as possible. Yeeyan is also considering to start biweekly addresses on its blog. Future steps may include opening a bug-tracking system to the community.

By doing these, Yeeyan is making its way toward a community-based enterprise.

COPYRIGHT-RELATED ISSUES

Copyright is an inevitable issue for open translation. It has been discussed and argued within the team since the first day of Yeeyan. By attending OTT09 (Open Translation Tools 2009) in Amsterdam, Yeeyan found its long-sought community and is willing to cooperate with Creative Commons to resolve copyright related issues and construct licensing platform on top of Yeeyan's UGC engine. This is the next big thing for Yeeyan.

TRANSLATING TEXT

- 27. Text Content
- 28. Web Translation Systems
- 29. Preparing Content
- 30. Translation Tips
- 31. Content Management Systems
- 32. Translating in a wiki environment

27. TEXT CONTENT

Whether you are translating a blog post, news article, or the transcription of audio or video content, translation almost always involves translating text from one language to another. Fundamentally the role of a translator is to read text from one language such as a news article in Chinese and write the same content in another language such as German or Luganda. Every translator and every translation community will eventually develop a workflow to accelerate and make more efficient the process of making content available in multiple languages.

While some translators work without using any tools the vast majority use at least one or more of the following.

TYPES OF TOOLS

1. **Dictionary** - A translation, or bilingual, dictionary lists suggested translations of individual words (and sometimes phrases) from one language to another. The largest and most popular open translation dictionaries are <http://wiktionary.org/>, <http://www.omegawiki.org>, and <http://open-dictionary.com/>
2. **Machine Translation** - Machine translation (MT) uses computers to translate text from one language to another. Machine translation is not available in all language pairs, and the resulting translations tend to be most accurate when working with languages from the same language family. Open source machine translation tools include <http://www.apertium.org/> and <http://www.statmt.org/monosmt/>.
3. **Glossary** - Unlike a dictionary which aims to define and translate every major word in a given language, a translation glossary or terminology list only includes special terms that require specific translations. Translators use glossaries when:
 - o the translation includes special terminology or jargon such as advanced scientific, medical, or technical texts.
 - o the translation is part of a larger set of documents that should maintain consistent terminology across all documents.

While a format like TBX exists for glossaries many tools use simple tab-delimited or comma separated files (CSV), the latter can be opened in a spreadsheet if needed. A collection of open glossaries organized by language is available at <http://www.lai.com/glossaries.html>

4. **Translation Memory** - As a translator works on an expanding body of documents it is likely that she will come across the same exact phrases over and over again. Each time a translation memory system comes across a phrase or group of words that have already been translated it will automatically replace the phrase with the previous translation. Open source translation memory systems include OmegaT (<http://www.omegat.org/>) and Anaphraseus (<http://anaphraseus.sourceforge.net/>). Google's Translate Toolkit (<http://translate.google.com/toolkit>) and Lingotek (<http://www.lingotek.com/>) are proprietary translation memory systems.
5. **Spell Check and Grammar Check** - Finally, many translators use spell checkers to help the editing process and to ensure consistency in the spelling of translated words. Most open source tools will use hunspell to provide spell checking. <http://jazzy.sourceforge.net/> is a server-based spell checker for projects hosted online.

TYPES OF WORKFLOWS

Ad Hoc Translation Workflows

Most translators use one or more of the above-listed tools in an ad hoc fashion. For example, an English to Bengali translator might use Anubadok (<http://anubadok.sourceforge.net/>) for machine translation, <http://www.ittefaq.com/dict/> as a dictionary and Ankur's Firefox plugin (<http://www.ankur.org.bd/>) as a spell checker.

Not every tool is available for every language. While there are no machine translation systems, for example, which translate between Malagasy and English, there is an online Malagasy dictionary with English and French translations (<http://malagasyworld.org/bins/alphaLists?lang=mg>).

Integrated Translation Workflow Systems

Professional translators tend to use integrated translation workflow systems which bring all five types of translation tools into one integrated interface. OmegaT and Virtaal, for example, are desktop application which integrate both translation memory and glossaries into a single translation workspace.

Though not open source, Google also offers a Translation Toolkit (<http://translate.google.com/toolkit>) which integrates machine translation, glossaries, translation memory, and dictionaries. (Explanatory video here - <http://www.youtube.com/watch?v=C7W2NjFdolg>) Trados (<http://www.trados.com/>) is another proprietary integrated translation workflow system.

Worldwide Lexicon (<http://www.worldwidexicon.org/>) is still in its early development stages but aims to offer an open source server-based integrated translation workflow system similar to Google's Translation Toolkit.

PUBLISHING AND ORGANIZING TRANSLATIONS

The most revolutionary aspect of the internet has been in enabling users to quickly publish and distribute content. In fact, the only three obstacles to potentially distributing content to every single human around the world are 1.) access to the internet, 2.) literacy, and 3.) language. Once a document has been translated there are various ways to link, integrate, and manage the translations. Creating links between translations of a text also creates links between the comments that follow in each language. Also, if the source text is corrected, updated, or edited then ideally those changes are also made to each of the translations.

Blogs / Ad Hoc Translation

The simplest solution is simply to invite your readers to translate posts and re-post them on their own blogs, and then cross-link to each other. It is best to publish your content using a license which allows for derivative works so that your readers know that they are free to produce and republish translations of what you write. For example, Kevin Kelly published an essay to his blog called "The Expansion of Ignorance."

(http://www.kk.org/thetechnium/archives/2008/10/the_expansion_o.php) Enzo Abbagliati, a Chilean blogger, then translated the post into Spanish, published it on his own blog, and left a comment on Kelly's blog post to make him aware of the translation (<http://abbagliati.blogspot.com/2008/10/la-expansin-de-la-ignorancia.html>). The essay was also translated by another reader into Chinese and Kelly later edited his post to point readers to all available translations. This method requires no specialized software, nor does it require that everyone use the same set of tools. You can publish your blog on WordPress, for example, while someone else translates the post on their Blogger-based site. This strategy works nicely for writers or blogs that have a following, and that publish on a light or occasional basis.

Multilingual CMS

A number of open content websites are now publishing their blog posts, news stories, and essays in multiple languages. Examples include <http://globalvoicesonline.org/>, <http://www.cafebabel.com>, <http://www.eurotopics.net>, <http://vocesbolivianas.org>, and <http://www.indymedia.org>.

Until recently, most content management systems had mediocre and unstable support for multilingual publishing. This is beginning to change as open source CMSs such as Drupal, Joomla, WordPress, Plone, and Tribiq now allow for the localization of the interface into multiple languages and multilingual publishing. These systems do not yet integrate with translation workflow systems, which must be managed separately, but they do enable you to give translators access to your publishing system so that they can publish, manage, and edit translations of content.

Social / Community Translation System

Social translation platforms encourage a community of translators to create, edit and curate translations from a variety of sources. Examples include Meedan (<http://meedan.net>), where volunteers translate discussions around current events in the Middle East into Arabic and English, and Yee-Yan, where volunteers translate news-related content into English and Chinese. Social translation systems tend to build a community of volunteer and professional translators around a specific topic, interest, or mission.

Wikis

Wikis are widely used for multilingual content projects like Wikipedia (<http://wikipedia.org>), which exists in over 250 languages, and WikiTravel (<http://wikitravel.org>), a free travel guide available in over a dozen languages.

Wikipedia/MediaWiki

Mediawiki, the software created by the Wikimedia Foundation (<http://wikimediafoundation.org/>) to run Wikipedia, now has improved multilingual support. Newer tools, such as TikiWiki, have designed the translation processes into the system itself. This is an area where we expect much growth and evolution over the next two to three years.

28. WEB TRANSLATION SYSTEMS

A web translation system is a web-based service that provides machine translation between two or more languages. Web translation systems rarely produce perfect translations (especially between languages that are not in the same language family), but they often help give readers a rough idea of the basic content of the text. Some translators use web translation as part of their workflow to either produce a draft version that they can edit into final translation or to create an alternate document to compare the translations of particular words and phrases.

Most content management systems offer plugins such as WordPress' Global Translator plugin [<http://wordpress.org/extend/plugins/global-translator/>] which offers machine translations in 41 different languages using Google Translate, Babel Fish, Promt, and FreeTranslations.

APERTIUM

Apertium (<http://www.apertium.org>) is a rule-based open source machine translation engine that focuses mostly on Latin-based European languages.

MOSES

Moses (<http://www.statmt.org/moses/>) is an open source statistical machine translation system that allows you to automatically train translation models for any language pair. Online versions of Moses currently translate between Czech↔English, English→Russian, Finnish↔{Swedish,English}, and English↔{German,Spanish,French}. (<http://www.statmt.org/moses/?n=Public.Demos>)

ANUBADOK

Anubadok (<http://bengalinux.sourceforge.net/cgi-bin/anubadok/index.pl>) is a free and open source machine translation system for English to Bengali translations.

GOOGLE TRANSLATE

Google Translate (www.google.com/translate) is not an open system but it is a powerful and free web translation service that enables users to quickly look up translations for a phrase, document or website. Google's system is a statistical machine translation system, which works by comparing parallel texts and their translations in many languages. With a sufficient large training set (typically millions or tens of millions of sentences), the system learns to translate texts by example. Like any machine translation system, Google's system does not provide human quality translations, but generally provides decent, approximate translations.

BABELFISH

Babelfish (<http://babelfish.yahoo.com/>), once owned by Alta Vista and now part of Yahoo!, is similar to Google Translate. It is powered by Systran rule-based machine translation.

TOGGLETEXT

ToggleText (<http://www.toggletext.com/>) is a proprietary web translator that focuses on English↔Indonesian translation.

OTHER SYSTEMS

29. PREPARING CONTENT

The amount of time it takes to translate a document and the quality of the resulting translation depend largely on how well the source text has been prepared for translation. By following a few basic best practices you can encourage the translation of your content, make the process more efficient, and ensure that the message is not lost in translation.

The translation industry employs a few strategies to help ensure that content can be well translated. These include:

- Constraining language - by limiting the terminology, complexity and style of technical manuals it is possible to ensure that they remain translatable.
- Pretranslation - in this process an editor, who understands the issues of translation into the target languages, makes changes to the source text to ensure that it is translatable.

COMMON PROBLEMS AND SOLUTIONS

The following are a list of the most common issues and how they might be addressed.

Style

The source content may be in various styles, some of which might not work in the target language. A simple example would be where content is in a very personal style while the target language employs a very impersonal style in this type of content.

The source content needs to be adapted to address the issue or the translation brief should specifically state the change in register is allowed in the target languages. In the long term it might be worth establishing a style guide for the source documents.

Complex Sentence

The creator of the source document might make use of a style that creates sentences with more than one key point. A pre-translation editor would break these into two sentences.

Consistent Use of Terminology and New Terms

It is always good to build a terminology list for the domain, this helps the translators when they are translating. In the same way the source document should consistently use that terminology. A pre-translation editor would adjust the use of terms to align with the terminology list.

Any new terms that are found that need definition and that will need to be developed in the target language are added to the terminology list.

Logical Flow of Arguments

In the heat of a blog post an author might make an argument that is poorly developed, that makes a leap of faith or that needs a minor tweak. A pre-translation editor would help to clarify this logic either by correcting it or adjusting it with the author. This ensures that translators are not faced with the issue of having to build the arguments themselves.

Repetition of Logic

An author may repeat the same idea a number of times using different examples or arguing from different directions to arrive at the same conclusion. A pre-translation editor would either merge these arguments into one, ensure that they are each logical or write something to the translators explaining that there are two points being developed.

Foreign Language in the Source Text

Content creators may include foreign phrases, borrowed words, slang and other words or expressions that the translator may not be familiar with. An English author writing in South Africa might borrow Afrikaans or Xhosa words and expressions. The pre-translation editor might remove these or explain their meaning in a general way so that translators can translate them. The editor could build the explanation into the source text so that it is easily translated and give instructions not to translate the original.

Content creators might want to avoid using terms that might be specific to their locale or to always explain words and phrases that could cause confusion. There is of course a balance in that a personal piece full of colour and expression should not become academic or plain.

Idioms, Examples and Cultural References

Idioms can be some of the hardest things to translate as they have many levels of meaning. A translator would need to understand those meanings to be able to find equivalents in their language. This is one reason why many people insist that translation be *toward* a translator's primary language as it is only in this language that the translator has full access to equivalents. A pre-translator can explain the idiom to the translator or even highlight the key part of the idiom that is being used in the context.

Examples are the easier of this group to adjust. It's often easy to find examples from the target language's locale. Thus, the pre-translation editor can either find general examples or allow translators to adjust the example to their locale as needed.

Cultural references would include quotes, movie dialogue, etc. "Play it again, Sam", "Open the podbay door, Hal", "Beam me up, Scotty" are all references to popular culture which may or may not be a part of popular culture in the target language. However, the target language might have a rich parallel popular culture. For example, science fiction culture in Hungarian is very rich thus offering alternatives. The pre-translation editor will choose their approach based on the target languages including asking for a similar reference, explaining the context of the reference or eliminating the reference.

30. TRANSLATION TIPS

First and most importantly, if you are thinking about volunteering to translate something you feel passionate about, enjoy yourself, don't forget why you are doing this. Translation is as much an art as it is a science, but there are some basic best practices that can help make translation easier and more rewarding.

UNDERSTAND WHAT IS TO BE TRANSLATED

- If you do not understand the subject or a sentence, you cannot translate it well by yourself. It is best then to leave it to others, translate only trivial (non technical or domain specific) material, or work with someone that is an expert in the topic.
- Understand the culture of the audience that you are trying to reach and think about the culture in the context of the original text.
- Understand the style the audience is used to. Avoid word-by word or "literal" translations, which always read awkwardly.
- Do you know the subject matter and understand its lingo? Ask if there is a glossary of terminology that is usually used by translators for this subject. If you do have a glossary, create your own to be consistent among your translations.
- Understand and maintain the formal/informal style of the source text.
- You must stay true to the original text, without reinterpreting it with your own ideas.
- Make sure that you understand the technical limitations and requirements of the translation that you want to do.
- It is always best to translate into your native language and extremely difficult to translate into a language that is not.
- Try to understand how long it may take you to translate this project, and think if you want to commit to it. Calculate how much time it will take. A page of text might take from 1 to 3 hours for a beginner or volunteer (1 or less for an experienced translator).

PREPARE YOURSELF

- Get the tools and resources that can make your translation more efficient and accurate: glossary, translation memory (TM), spell-checker, dictionaries, thesaurus, encyclopedias.
- Read the instructions very carefully (if provided). If there are no instructions, feel confident to ask the author of the source text questions.

WHILE YOU DO IT...

- After you write a few sentences, or a paragraph, read it to make sure it makes sense.
- Don't over do it, there is no such thing as a perfect translation, but you do of course want to create a good translation
- Use a spell-checker, track changes (this is a feature of some Word Processors and online tools) or any other tool that you have access to.

AND DON'T FORGET TO:

- Ask permission to translate an original text if it is not published under a license which explicitly allows for derivative works.
- Don't be afraid to ask for attribution/credit for your work.
- Look at the license.
- Think of the author(s) and how you should treat the text and use it in a polite way.

31. CONTENT MANAGEMENT SYSTEMS

Content Management Systems (CMS) were designed to make it easier for people to manage complex websites with multiple authors, editors and translators, each with different roles and privileges within the system. These systems have been around since the beginning of the web itself, and have different features depending on the type of system and its intended use. Content management systems have recently begun to recognize the importance of multilingual publishing, and now offer ways to publish and manage articles in more than one language.

Because of this publishers take several different approaches when building multilingual sites. Among the strategies they use are:

- Adding automated links to machine translation versions of your content.
- Building separate, parallel sites for each individual language.
- Using a single, multilingual CMS (e.g. Drupal, Joomla, WordPress) to manage translations of original articles.
- Using an RSS translator to import content to a platform where entries can be translated into other languages. Translations are then made available in a separate RSS feed.

Which approach is best depends on the type of site you run, how many articles you post per day, and how important it is for multilingual content to be integrated with your primary website or language.

SETTING UP A PARALLEL SITE

If you are building a fairly simple site, with a relatively low volume of posts, and expect other languages to be a minority of your audience, the best approach is to set up a parallel, sister site for each language you seek to support. For example Yoani Sánchez is a popular Cuban blogger who writes in Spanish at <http://www.desdecuba.com/generaciony/>. Many of her readers have volunteered to translate her blog posts into over a dozen other languages. Those translations are published on parallel blogs. For example, Japanese translations of Yoani's posts are available at http://desdecuba.com/generaciony_jp/.

This approach requires minimal additional effort and is easy to manage.

Some tips for success with this method include:

- Cross-link source and translated entries.
- Use multilingual tags for posts, so search engines are more likely to pick up on foreign language search terms.
- Embed block quotes from the original in the translation, both so readers can compare the original text, and so that search engines are likely to work for both original and translated search terms.
- Encourage translators to contact you, so you can maintain an index of translations and translators on your site, and encourage them to continue doing so.

ADDING MACHINE TRANSLATION TO YOUR SITE

Machine translation can offer readers a general idea about the content of articles, but accuracy is often a problem, so this is not a good option if you require accurate translations. Google Translate offers a number of useful widgets, web badges and other tools you can embed in your site to enable users to quickly auto-translate the pages or sections they are interested in. Most content management systems offer plugins which will embed links to machine translations in various languages to all articles. For example, Drupal has an optional module which will add links to machine translations to all content on a Drupal-based website.

<http://drupal.org/project/gtrans>)

MULTILINGUAL CONTENT MANAGEMENT SYSTEMS

Another approach is to use a multilingual CMS, such as Drupal, Joomla, or WordPress. These systems allow a single post to be associated with translations, which in turn can be displayed adjacent to the original post, or when the user switches to a different language using a language selection menu on the home page.

These translations are not generated automatically, so you need people to create them, and you need to have a system for notifying translators of new articles and changes to articles which have already been translated.

Some CMSs, Drupal especially, are designed to support translations in the core system. This means you can "skin" a post in any number of languages. You might write the original post in English, and then create many child documents in other languages, which the system treats as translations of the original post. So when the user switches the user interface to German, he will see German versions of posts and pages, if they are available.

SAMPLE MULTILINGUAL PUBLISHING WORKFLOW

There are many content management systems and many different plugins for each system that allow you to publish, manage, and organize content in multiple languages. What follows is a sample workflow using WordPress (<http://wordpress.org>) as a content management system and WPML (<http://wpml.org/>) as a plugin to manage translations. A guide to WordPress is available at <http://en.flossmanuals.net/wordpress>.

First you must publish the original article in its source language:

Edit Post

[Podcast] A Latin Indie Techno Hipster at Heart

Permalink: <http://el-oso.net/blog/archives/2009/06/09/podcast-a-latin-indie-techno-hipster-at-heart/> [Edit](#) [View Post](#)

Upload/Insert

[b](#) [i](#) [link](#) [b-quote](#) [del](#) [ins](#) [img](#) [ul](#) [ol](#) [li](#) [code](#) [more](#) [lookup](#) [close tags](#)

`modismos` I learn, no matter how many lyrics I can sing, no matter how many dishes I can cook. But walking through the streets of Latin American cities and pueblos always fills me with a sense of familiarity, a sense of calm.

Today's podcast is a collection of some of my favorite electronic music from Latin America. It is meant for long walks through cobblestoned `callejones` of your favorite Latin American city.

[podcast]<http://el-oso.net/mp3/Latin%20American%20Electronica.mp3>[/podcast]

`Download (Right-click, save as)`

Word count: 331 Last edited by oso on Tuesday June 9th, 2009 at 8:29 am

Language options

Language **English**

Not translated

Translations ([hide](#))

Language	Title
Spanish	n/a

Operations [add](#)

Once the post is published in its original language you then must click on "add" to add a translation of the post in another language. When you click "add" you will be redirected to another "Add New Post" page:

Add New Post

Un Indi-Tecno-Hispter Latino de Corazón

Permalink: <http://el-oso.net/blog/archives/2009/06/29/un-indi-tecno-...ino-de-corazon/> [Edit](#)

Upload/Insert

[b](#) [i](#) [link](#) [b-quote](#) [del](#) [ins](#) [img](#) [ul](#) [ol](#) [li](#) [code](#) [more](#) [lookup](#) [close tags](#)

``

La frase escrita en la pared al otro lado de la calle dice:

`<blockquote>Ni indigenismo, ni oligarquía
Viva la revolución obrera</blockquote>`

La calle principal, de concreto arrugado y lleno de huecos, se ramifica en un vasto laberinto tejido de callejones empedrados, bautizados con los nombre de países Latinoamericanos, líderes de esta revolución o la otra. Uno de cada diez edificios es un esqueleto de concreto y ladrillo, una

Word count: 357 Draft Saved at 5:52:35 am.

Language options

Language Spanish

Translations [\(hide\)](#)

Language	Title	Operations
English	[Podcast] A Latin Indie Techno Hipster at Heart	View

In addition to translating all of the text of the blog post, it is also important to translate some of the metadata in the HTML tags. The ALT tags of all images should be translated. At times it is also appropriate to change links to resource information. For example, bloggers often link to Wikipedia articles to provide more background information. It usually makes sense to change the link to a Wikipedia entry in the same language as the translation.

When you publish the translation an automatic link will be made between the original article and the translation. Optionally, WPML can automatically detect the language of the browser of your readers and direct them to content in their language.

RSS TRANSLATION

For sites that want to automate the translation process, you can use RSS to enable this. The basic concept is to mirror your content onto a translation hub or translation management system, where translators go to view, create and edit translations. From there the translations can be imported back into your publishing system via translated RSS feeds. This technique enables you to integrate translations into your website without making major changes to your CMS.

Machine Translation and RSS

One thing you can do is to use machine translation to obtain fast but inaccurate translations for everything you publish. The Worldwide Lexicon RSS translator is an easy way to do this. Simply deploy an instance of WWL, and add your site's RSS feed to the index of sites the RSS translator monitors. It will automatically pick up and translate whatever you publish to approximately 40 languages. You then direct your visitors to go to www.yourtranslationhub.com, where they will see your articles, blog posts etc in translation. They can also edit the machine translations simply by clicking on a paragraph of translated text.

Human Translation and RSS

By mirroring your content in to a translation hub, users can easily submit, edit and curate translations for your content. You can either host the translations on the translation hub, or you can import translations back into your site via translated RSS feeds. For example, you can set up translate versions of your site at www.yoursite.com/es and www.yoursite.com/fr for Spanish and French, with each importing content from www.yourtranslationhub.com/rss/es and www.yourtranslationhub.com/fr respectively. This approach requires more work, but it enables you to display translations natively within the context of your parent website. You can either publish the imported articles automatically (faster), or store translated posts as drafts/unpublished so that they are reviewed by one of your editors prior to publication.

32. TRANSLATING IN A WIKI

ENVIRONMENT

If you are translating content from another wiki, consider whether you need to do a full translation or whether a partial translation or a summary is more appropriate. In some cases an adaptation may be as good as or better as a full translation and may take a lot less time.

If you are translating an article in order to introduce the article or a piece of it as new content into your wiki, you don't have to get it all done at once. Translate a few choice paragraphs, or the intro and the lede, and if you need to take a break, save and take a break. Translation, just like content creation, is a collaborative process on a wiki; take advantage of it!

It is important to keep in mind that changes in most wikis go live immediately. Although your translation need not be perfect, it is bad form to translate for speed and leave the cleanup to others, unless you check in with them first.

Unlike most commercial translation projects, there are generally no instructions provided when you sit down to translate a wiki article. If you run across something ambiguous - a specialised term that you don't know, or even something wrong in the original text - it may be difficult or impossible to track down the author(s) and get clarification from them. Certain information you would expect to see in written instructions will have to be deduced from the article itself, including: the background knowledge of the audience, appropriate style and tone, and so on.

If you are going to be working on a long article, tag it in some fashion so that others know you are working on it. If you translate slowly, that's no problem; mark a few sections as in progress and other translators can work on other sections.

The source text and the original authors must be referenced somehow from the translated content if the license of the source text requires it. Most open content licenses (the GFDL and the CC-BY licenses for example) require this. Check to see how your wiki and the source wiki handle attribution. You may be able to get by with a link to the source page and with the mention of the primary authors in the edit summary at the time you create the translated page, or special procedures may be necessary.

Wikis often do not have a mechanism in place for a translate-proof workflow. In those instances, when possible, it's good to create out-of-band mechanisms to implement a workflow. Even untranslated text gets edited and proofread by people other than the original author before being published. In your case, the content may go live immediately, but you can still work out an arrangement ahead of time for someone else to proofread shortly after you submit. This will not only improve the quality of the wiki's content but will contribute towards building of a community of translators on your wiki.

Talk to other translators on your wiki: for advice, for moral support, and for fun. Professional translators have community fora; why shouldn't the rest of us? If translation is going to be more than drudgery for you (and we hope it will be), you'll want to share the joys and quirks of it with others doing similar work.

TRANSLATING VIDEO

33. Subtitles

34. FILE FORMATS

35. Finding Subtitles

36. Creating Subtitles

37. Playing Multilingual Video

38. Distribution

33. SUBTITLES

Subtitles are generally text translations of the source language of the video that show up on screen. They allow videos to be translated into any language that has an available script, called character set, and thus can potentially have a global viewership.



Photo courtesy of Antoniot78 on Flickr (Creative Commons License)

Subtitles come in a few file formats and can be attached to video in a few different ways. This variety can give subtitled video a greater flexibility but at the same time less standardization can also create headaches. However, the basic construction of a subtitle is a block of text linked to a time code that matches a certain point of time within the video. During video playback, when that point happens in the video, the subtitle also appears.

Captions are another type of text overlay for video content. Captions are used mainly for accessibility purposes - for deaf or hard of hearing people. Captioning is used to describe a wider range of information than subtitles, for example descriptions of non-spoken events such as noise, music and dramatic events. See this article by Joe Clark for more information about online captioning - <http://joelclark.org/access/captioning/bpoc/ST.html>

34. FILE FORMATS

A subtitle file format specifies the format of a file (text or image) containing the subtitle and timing information. Some text-based formats also allow for specifying styling information, such as colours or location of the subtitle.

Some subtitle file formats are:

1. Micro DVD (.sub) - a text-based format, with video frame timing, and no text styling
2. Sub Rip (.srt) - a text-based format, with video duration timing, and no text styling
3. VOB Sub (.sub, .idx) - an image-based format, generally used in DVDs
4. Sub Station Alpha / Advanced Sub Station (.ssa, .ass) - a text-based format, with video duration timing, and text styling and metadata information attributes.
5. Sub Viewer (.sub) - a text-based format, with video duration timing, text styling and metadata information attributes.

EXAMPLES

Lets look at the actual content of some subtitle files. They will all be simply showing "This is my first subtitle!" in the first 10 seconds of video playback. These were all produced by the FOSS subtitling software Jubler.

The first thing to note is that each file is simply a text file, and is editable by any text editor, such as vi on GNU/Linux, or Text Edit on Mac, or Notepad on Windows.

The following is how our example is realised in a Micro DVD subtitle file (presuming 25 frames per second):

```
{0}{250}This is my first subtitle!
```

As a Sub Rip subtitle file:

```
1
00:00:00,000 --> 00:00:10,000
This is my first subtitle!
```

As a Sub Station Alpha (.ssa) file:

```
[Script Info]
; Edited with Jubler subtitle editor
Title:
Original Script: andycat
Update Details:
ScriptType: v4.00
Collisions: Normal
PlayResX: 320
PlayResY: 288
PlayDepth: 0
Timer: 100,0000

[V4 Styles]
Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, TertiaryColour, BackColour,
Bold, Italic, BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel,
Encoding
Style: Default,Arial Unicode
MS,31,&HFFFFFF,&H00FFFF,&H000000,&H404040,0,0,1,0,2,2,20,20,20,255,0

[Events]
Format: Marked, Start, End, Style, Name, MarginL, MarginR, MarginV, Effect, Text
Dialogue: 0,0:00:00.00,0:00:10.00,*Default,,0000,0000,0000,,This is my first subtitle!
```

As an Advanced Sub Station (.ass):

```
[Script Info]
; Edited with Jubler subtitle editor
Title:
```

Original Script: andycat
Update Details:
ScriptType: v4.00+
Collisions: Normal
PlayResX: 320
PlayResY: 288
PlayDepth: 0
Timer: 100,0000

[V4+ Styles]

Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, OutlineColour, BackColour, Bold, Italic, Underline, StrikeOut, ScaleX, ScaleY, Spacing, Angle, BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, Encoding
Style: Default,Arial Unicode
MS,31,&H00FFFFFF,&H0000FFFF,&H4B000000,&H4B404040,0,0,0,0,100,100,0,0,1,0,2,2,20,20,20,0

[Events]

Format: Layer, Start, End, Style, Name, MarginL, MarginR, MarginV, Effect, Text
Dialogue: 0,0:00:00.00,0:00:10.00,*Default,,0000,0000,0000,,This is my first subtitle!

As a Sub Viewer (.sub) file:

[INFORMATION]
[TITLE]
[AUTHOR]andycat
[SOURCE]
[FILEPATH]
[DELAY]0
[COMMENT]Edited with Jubler subtitle editor
[END INFORMATION]
[SUBTITLE]
[COLF]&HFFFFFF,[STYLE]bd,[SIZE]18,[FONT]Arial
00:00:00.00,00:00:10.00
This is my first subtitle!

There are large numbers of file formats around (see

<http://diveintomark.org/archives/2009/01/07/give-part-4-captioning> - the main ones mentioned by this article not covered here are MPEG4 Timed Text, SMIL and SAMI).

COMPARISONS OF FORMATS

Tables of comparisons of subtitles file formats are found at the following:___

<http://www.annodex.net/node/8>

<http://en.wikipedia.org/wiki/Subtitles>

SUPPORTED FILE FORMATS IN FLOSS VIDEO PLAYERS

A list of subtitles supported by the FLOSS video player, VLC, can be found at:

<http://wiki.videolan.org/Subtitles>

35. FINDING SUBTITLES

For some subtitle translation, pre-made subtitles may be a useful resource particularly if the video is a well-known or commercial work. For example, if you are including a scene from an American documentary in a video, there are resources to search for existing subtitles in a given language. However, outside of well-known video and cinema, pre-created subtitle resources are few and open source resources are even fewer. When they *do* exist, they come in the form of open source corpora and translation memories. Both are a type of repository for parallel translated language phrases and segments. Subtitles are then able to be translated with a search and find technique. This can be an especially useful tool for translating idiomatic expressions and common word strings.

There are a few issues that come up when searching for subtitles. For cinematic films, for example, there are almost invariably many different versions of the film. One can imagine that any extra scene, extended title sequence or formatting change can alter the timing of subtitles onscreen which many times renders subtitles useless. Therefore, it is important to find subtitles that are accurate for the audio of the particular film version. There are tools like the open source Sub Downloader (<http://www.subdownloader.net/>) that help with this problem by matching subtitle sets to specific film versions. Another issue that comes up is the file format of the subtitle file itself. There are different formats for different types of video as well as different types of physical media (HD, DVD, Blu Ray etc.) which affect the selection of subtitles for a given piece of film. In short, details about the film and audio change the availability of subtitle resources.

Resources:

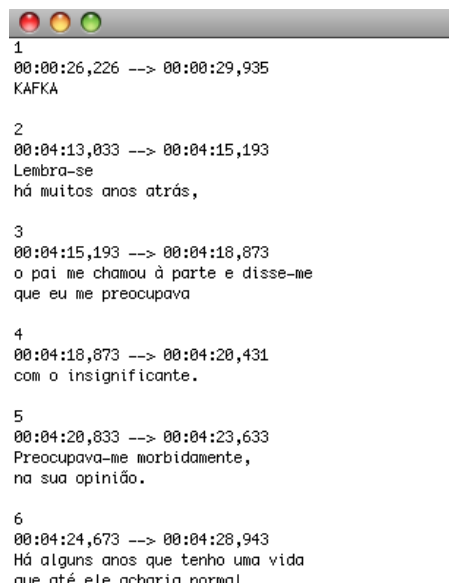
- OpenSubtitles.org: <http://www.opensubtitles.org/en>
- TinyTM: <http://tinytm.sourceforge.net/>
- DivX Subtitles: <http://www.divxsubtitles.net/>
- AllSubs.org <http://www.allsubs.org>

36. CREATING SUBTITLES

Subtitles files can be created by text editors, or more specialised software like Jubler, GnomeSubtitle, Gaupol and SubtitleEditor. Lets look into a specific example of a subtitle file, and open it in a text editor (eg Text Edit on MacOS, Notepad on Windows or GEdit on GNU/Linux) and modify the subtitles to see it change in video playback.

The screenshot below shows Text Edit on Mac OS X with a Portuguese Brazilian translation in Sub Rip (.srt) format for the movie *Kafka*. You can find this translation :

<http://www.opensubtitles.org/en/subtitles/3506361/kafka-pb>



```
1
00:00:26,226 --> 00:00:29,935
KAFKA

2
00:04:13,033 --> 00:04:15,193
Lembra-se
há muitos anos atrás,

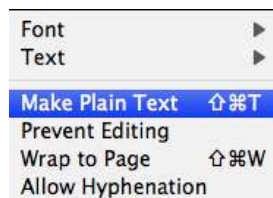
3
00:04:15,193 --> 00:04:18,873
o pai me chamou à parte e disse-me
que eu me preocupava

4
00:04:18,873 --> 00:04:20,431
com o insignificante.

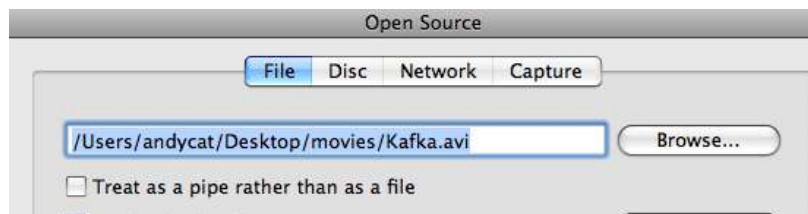
5
00:04:20,833 --> 00:04:23,633
Preocupava-me morbidamente,
na sua opinião.

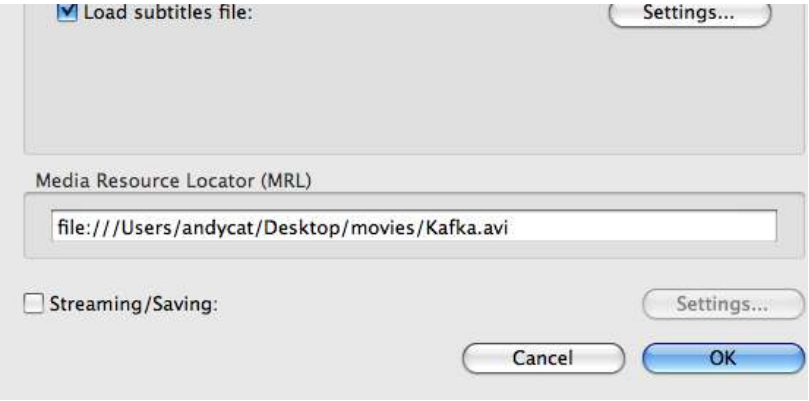
6
00:04:24,673 --> 00:04:28,943
Há alguns anos que tenho uma vida
que até ele acharia normal
```

As a side note, in TextEdit, remember you need to be in 'Plain Text' mode to edit SRT files. Go to Format -> Make Plain Text, if you happen to be in Rich Text mode, as show below:

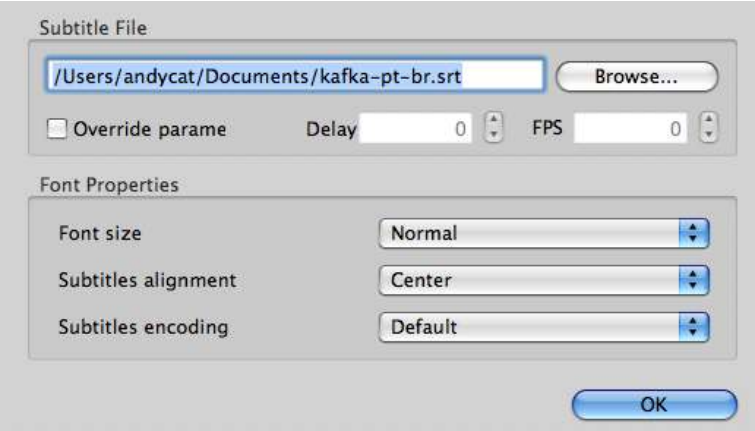


Using VLC (an Open Source media player), I can start Kafka and load this subtitle, as show below:

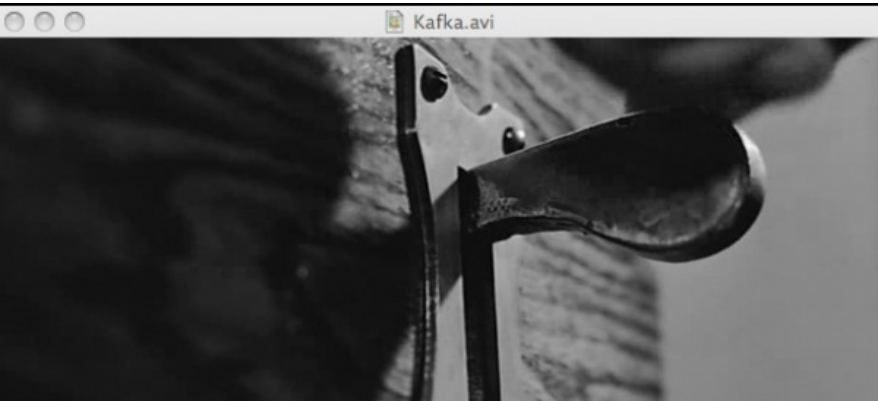


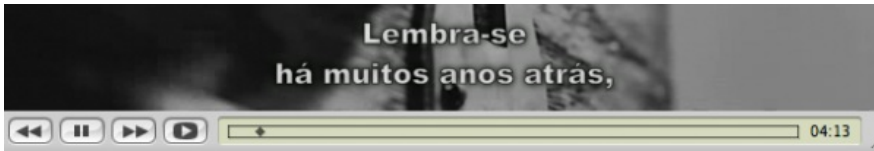


Remembering to load the subtitle file associated with it. Note it could be in a different location, or named differently from what is shown below:



As you can see in the above screenshot of the SRT file, the first real dialogue is approximately at 00:04:13 in hh:mm:ss format. That is 4 minutes 13 seconds. We can see this subtitle in the video window of VLC, as shown below:





Now, let's return to our text editor, and make some changes to the file to show how easy it is to create and/or modify subtitles.

Let's change this text to 'This is my first subtitle!' just as an example. Here is the modified, and saved, subtitle file.

```
1
00:00:26,226 --> 00:00:29,935
KAFKA

2
00:04:13,033 --> 00:04:15,193
This is my first subtitle!

3
00:04:15,193 --> 00:04:18,873
o pai me chamou à parte e disse-me
que eu me preocupava

4
00:04:18,873 --> 00:04:20,431
com o insignificante.

5
00:04:20,833 --> 00:04:23,633
Preocupava-me morbidamente,
na sua opinião.

6
00:04:24,673 --> 00:04:28,943
Há alguns anos que tenho uma vida
que até ele acharia normal.
```

Now, replaying the Kafka video with the subtitle shows:





The above shows how easy it is to manually edit subtitles within a simple text editor. We have not show any time code modifications, nor have gone into file format specifics. You should know the details of the file format you are manually editing if you want to go further into hand crafting subtitle files.

To go further with subtitle production, we need to start to investigate specific subtitle editing software.

37. PLAYING MULTILINGUAL VIDEO

Software name : VLC

Software version : 0.8.6

I assume you have VLC player installed and you have a file or DVD with subtitles which you want to display when you are playing the Video.

There are three ways you may want to use VLC to display subtitles.

- 1) From a DVD
- 2) From a Multilingual file (ie Matroska)
- 3) From a separate subtitle file which is distributed with the Video file.

PLAY SUBTITLES ON A DVD DISK

To do this put the DVD disk into your DVD drive. Open up VLC player and select File > Open Disk.



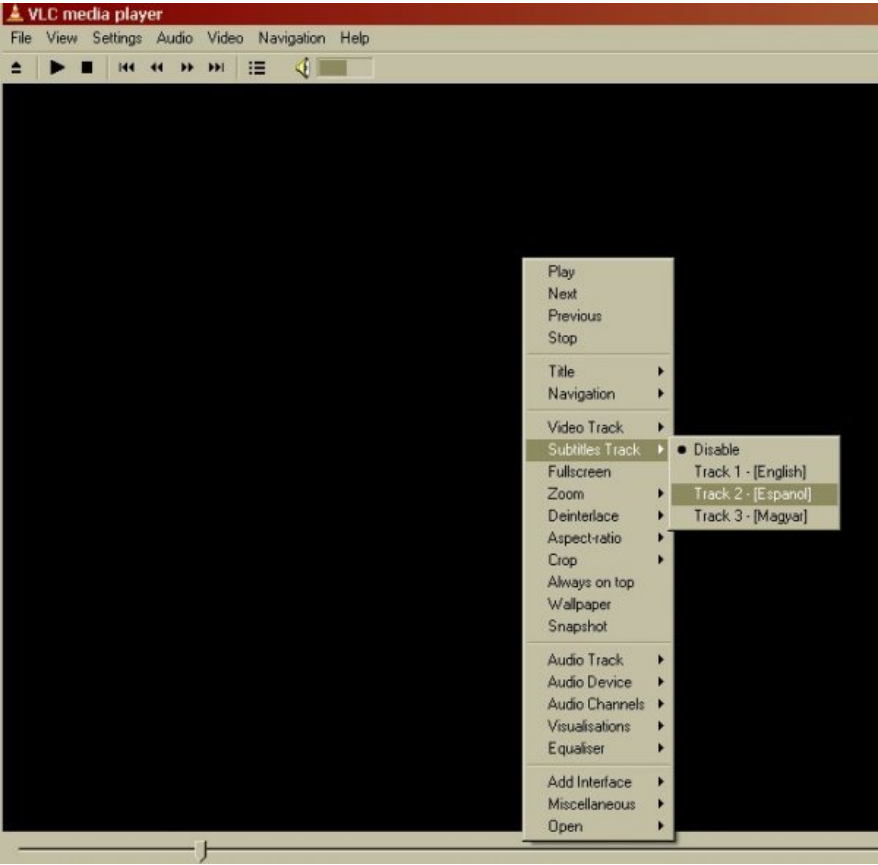
Enter the DVD Drive letter. It may appear automatically. Mine is drive D:



Then click OK.

The menu page of your DVD should appear.

Click on the video you want to watch. Then when the video starts quickly right hand click the mouse on the Video image. Select the Subtitle track you wish to view.

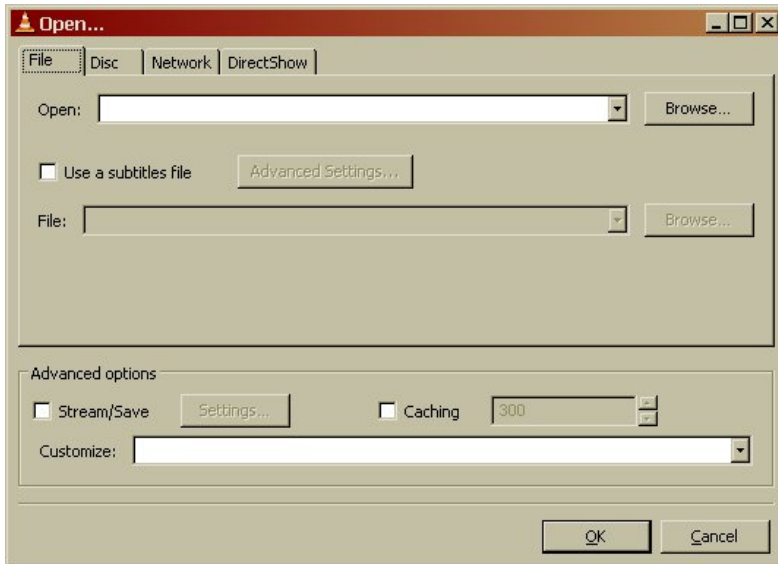


The subtitles should then appear on screen.

PLAY SUBTITLES IN MATROSKA FILES

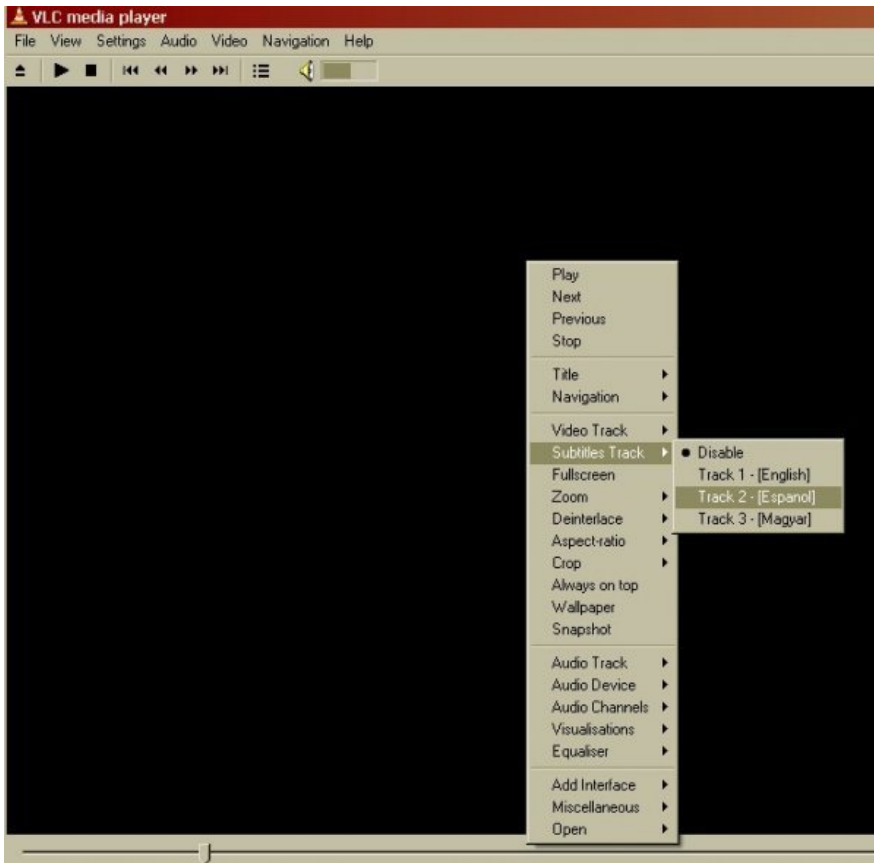
The process for this is exactly the same as above except when starting the process you select File > Open File

You then see this screen.



You should then click on the Browse button to select the video file you want to play. If this file is a matroska file with an *.mkv extension then you can click OK after browsing for the file as the file already has the subtitle information.

Then Select the subtitle language stream by right clicking the video screen and selecting Subtitle Track > and choose the language

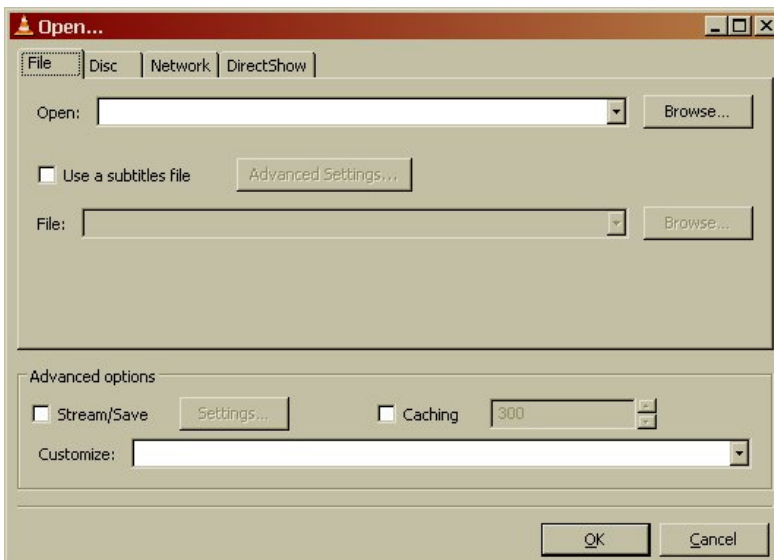


PLAY VIDEO FILES WITH EXTERNAL SUBTITLES

Using VLC to play Video file with external subtitle files.

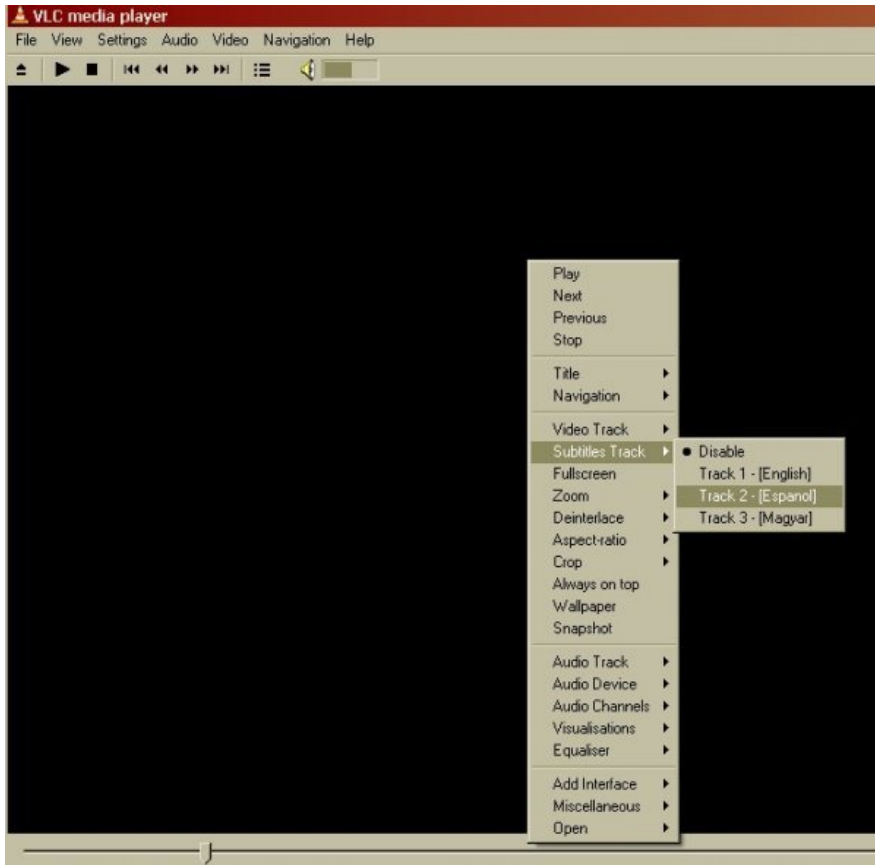
If you want to play an external subtitle file for example a srt file.

Select File > Open File



In the **Open** box click the **Browse** button and choose your video file.

Then put a tick in the box **Use a subtitle file**, and click Browse to locate your external subtitle file.



Then Select the subtitle language stream by right clicking the video screen and selecting Subtitle Track > and select the track of subtitles (for an external file like an srt file there will normally only be one track).

38. DISTRIBUTION

Video translation through subtitles is largely useless if the media cannot be distributed. There are many issues that come up when considering how to distribute subtitled video. First, file format differences and preferences can affect the accessibility of your content. Second, the method of distribution, actually how the video is sent out. Third, the resting place or home of the video content is important. Lastly, the license and re-usability of the content must be considered. All of these topics are dependent both on the intentions for the video and the audience which can change significantly from project to project. Therefore, some basic definitions and concepts are explained for further explanation and exploration of the options available.

You can choose to burn in the subtitles onto the video, ie have video editing software permanently render the subtitle text, at the correct times as indicated by the subtitle file, over the top of the video image. This means the video can be distributed as only one file, and the users dont need to worry about separate subtitle files and enabling subtitles in their players. However, you cannot get rid of the subtitles from this video, and need to produce separate video files for every translation you have.

On the other hand, you can simply produce separate subtitle files for every language which gives you and your audience extra flexibility. You need only distribute one version of your video, however now you will need to distribute subtitles for multiple languages, generally available as separate downloads.

Its also possible to explore the video container formats that allow embedding subtitles within the container, which provides the best of both worlds described above - the ability to not show subtitles, or one choosen from among the translations you make available, all within one file. Patent-unencumbered copyleft video container formats that support this include Matroska Multimedia Container (MKV) and the Ogg container format.

Lets briefly describe the tools you would use to render or distribute the subtitles you produce for your video. Avidemux, a FLOSS video editor, allows you to render subtitles over a video, and re-export this video with the text permanently embedded into the video. For distribution of web video, you can combine certain FLOSS video players, such as Flowplayer, with SRT files for embedding your video into a web page, and allow users to see subtitles render over the top of the video. At the cutting edge, you can experiment with the new <video> tag of HTML5 for playback of Ogg Theora video with Firefox 3.5 , using Java Script to control playback of the captions from SRT files.

ATTACHING SUBTITLES TO VIDEO

There are a few ways in which subtitles can be associated with the media they are translating:

- Including Subtitles as a Separate File

Allowing more flexibility for the media and subtitles as a whole, including the subtitles as a separate files allows that file to be accessed, changed and even taken out without affecting the video file itself. The disadvantage of this technique is that the subtitle file format becomes an issue. Players must accept the format in order to properly display the subtitles.

- Burning Subtitles into the Video Itself

Attaching the subtitles to the actual video "burns in" the subtitles with no separate file needed. This allows for a more universally playable subtitled video. However, at the same time, the video will then not be able to play without the subtitles and the subtitles themselves will not be editable or accessed. They are, for all intents and purposes, part of the video picture itself.

- Multiple Language Streams within a Single Video File

An emerging open source video technology is grouping different language subtitles into the one video container. These are multimedia container format streams that keep various streams of data separate within a single multimedia file. Just as audio and video are separate streams within a single movie file, this technique places the different subtitled languages together within the file allowing both increased playability and language choice for a single video. The Ogg and Matroska container format both support embedding various subtitle formats along with the video and audio streams, encapsulated within one digital file.

DISSEMINATING SUBTITLED VIDEO ONLINE

Fortunately, there are many FLOSS video hosting services that allow people around the world to see their videos and find out about the information important to them. One of the advantages of using an FLOSS hosting service is that both the multimedia content and the software platform is open to community contribution and collaboration. As with most web services, each has its own flavor and advantages.

- **EngageMedia**

<http://engagemedia.org>

EngageMedia.org is a video sharing site centered around social justice and environmental issues in the Asia Pacific region. They aim to create an online community based on open video. Video uploads are focused on social justice and the Asia Pacific. The not-for-profit Engage Media collective sponsor the software development of Plumi - <http://plumi.org/> - a FLOSS video sharing platform, which is the software that powers engagemedia.org. EngageMedia have also produce a guide for activists and NGOs on distributing video online - <http://www.engagemedia.org/guide-to-digital-video-distro>

- **Archive.org**

<http://www.archive.org>

The Internet Archive is an American nonprofit that seeks to provide public access to historical content and media in the form of a library online. Content uploaded and accessible includes text, audio, images, video and software. Valuing the preservation of historical content, this content highlights cultural and national milestones and the everyday. Registered users can upload content including video if they own the rights.

- **V2V.cc**

<http://v2v.cc>

V2V.cc is a user-submitted video content website that focuses on the ShareAlike Creative Commons license which allows you to "to copy, distribute, display, perform and derivative the content as long as it will be specified by the author and put under the license identical to [the existing] one."

- **Miro**

<http://www.getmiro.com>

Miro is an internet video application that allows you to view television from the internet through a downloaded application. Open source and free, Miro encourages community contributions both in the form of development to the platform and video content. When publishing, Miro assigns a video RSS feed so that users around the world can "subscribe" to your channel being notified when new content is added to your channel.

TORRENTS

Another option for distributing video content online is the use of BitTorrent. This method allows a user to upload video (or other content) onto a network, in effect "seeding" the network with the content. Peers on the network are then able to download the content and can become seeds themselves. In this way, the information for a large amount of content is spread out across a large number of users as more and more people download the content. The more people that then have the content and offer it as a seed, the more likely a successful and quick download will occur. This method is best for content that needs an efficient method of distribution rather than publicity and visual exposure.

TRANSLATING IMAGES

39. Introduction

40. What is an Image?

41. Image Tools

42. What is an SVG

43. Translating SVG

44. Scripting SVG

39. INTRODUCTION

Translating images can be a nasty task if they contain text, the image background is complex or the original image is of poor quality or resolution. While best practices for translating and creating new images can be an expansive subject, there are fortunately many free software programs that assist the image translator with image creation and editing.

In the Free Software world, there are a few standard image editing software programs. When working with image formats such as jpeg, png, or gif, GIMP is the de facto editor. Notably only PNG is a free (libre) format, although gif is now free in most countries. If you work with vector graphics the most sophisticated Free Software tool is Inkscape. Both GIMP and Inkscape are available for Linux, Macintosh and Windows.

Using these tools, one has much flexibility for editing images however it is important to recognize the possibility of others localizing the image. Separating text from the image makes editing substantially easier. In many tools, text can be added as a separate *layer*, which means that the text can be edited and changed without affecting the rest of the image. However, layers can only be changed if you have the original source files that preserve them. Images in jpeg, gif, and png do not preserve layers. GIMP's source file is in XCF format. While those working in Photoshop will save their source files as PSD files. GIMP is able to import the native and proprietary format for PSD files using a plugin with CMYK colorspace support :

<http://www.blackfiveservices.co.uk/separate.shtml>

In either case, these source files must somehow be made available to translators. Most files used in word processors, presentations, or in web pages are png, jpeg, or gif as these formats are best suited for display in a web browser. The file size of XCF and PSD files, on the other hand, are too large to put on the web and browsers do not render them. So a strategy is necessary to make the source files of images not only available, but also easily identifiable.

One strategy for making images easily translated is to use the open standard SVG, a vector graphic format. Text in an SVG image can be changed without changing the rest of the image because an SVG file is itself a text file. While SVG files can be edited in a text editor, there are SVG editors like the open source tool Inkscape that make it much easier. One of the main advantages of SVG is that the image file itself is the source file. However, SVG as a format is not widely supported by many applications, and most importantly not all browsers will display SVG. Therefore, many times it's necessary to export SVG to png (or another supported format) to use it in a webpage and are left again with the problem of keeping the source SVG file somewhere available for translators.

40. WHAT IS AN IMAGE?

You will most likely be familiar with digital images through webpages or the pictures on your computer. Digital images are produced by scanners, digital cameras, image software (eg. Paintshop, GIMP, Adobe Photoshop) or other devices and software.

A **Digital Image File** is a file that contains the information that describes the image. This file describes columns and rows which make up the image. Each unit of these columns and rows is known as a **pixel** (short for **picture element**). A pixel is the smallest unit that makes up an image.

FORMATS

There are many methods for describing how the pixels and columns and rows make up an image. The way you describe how these components work together is known as the **format**. For example, a digital picture of a house can be described in many ways. You may decide to use the method for describing the picture known as **JPEG** - so you save the picture as a **JPEG**. This means that the image file describes the image of the house using the rules of the **JPEG** format. If you saved the picture as a **GIF** then the rules describing the picture of the house conform to the **GIF** format. Each of these formats has its strengths and weaknesses depending on the purpose. The most used formats are **TIFF**, **PNG**, **GIF**, and **JPEG**. It is good to know a little about image formats so you can know which is the best format for your purpose.

FILE COMPRESSION

File size is an important function of a format. Some formats reduce the size of the file dramatically, this can aid the delivery of images over the internet where file size is a factor in determining how fast an image loads in a browser. The process of making a file smaller is referred to "as compressing the file".

When you want to make a file smaller you can compress it in one of two ways, to visualise this you can imagine placing a object into a plastic bag. How do you make the volume of the plastic bag smaller? You can either throw stuff away, or you can make the bag fit better around its contents (by sucking the air out for example).

If you throw stuff away you are losing some of the original contents. You can keep doing this until there is nothing left, reducing the bag volume more and more until nothing is left.

If you suck the air out of the bag you have the same contents but the total size of the bag is smaller. However you can only do this to a certain point - if you had an apple in the bag, for example, you could not get the bag to be smaller than the apple.

These strategies apply to image files too. If you throw data out of the image file you reduce the size but you also are losing data which means the quality will be reduced. This is known as **lossy compression**. If you optimise the file size by 'compacting' the data without throwing data away you reduce the file size but only to a certain point. This is known as **lossless compression**.

Different file formats approach compression in their own way. PNG and TIFF use lossless compression so are called **lossless formats**, while JPEG and GIF are **lossy formats**.

PNG

PNG is short for **Portable Network Graphics**. It was created as a royalty-free replacement for GIF. PNG uses lossless compression strategies, relying heavily on using smart mathematical ways of finding and describing patterns in an image file. Hence if you have an image with a large area of the same color then PNG is very effective in reducing the file size. PNG is also the only lossless format supported by browsers so if you wish to display images online without losing quality then PNG is the format for you.

PNG supports transparency which is good for creating fades, placing images nicely on a webpage regardless of what the background color is etc. However while this will look good in Firefox (for example) Internet Explorer 6 (and earlier) does not support transparency in PNG so transparent parts of a PNG image will display as grey in Internet Explorer in these browsers. Internet Explorer 7 displays PNG transparency correctly.

PNG does not support animation so you cannot make moving images similar to what you may have seen with **Animated GIF**.

When saving an image in the PNG format you can use the suffix '.PNG' or '.png'.

TIFF

Once known as **Tag Image File Format**, TIFF has been around for a long time, and the format is now owned by Adobe Systems Incorporated (the manufacturers of Adobe Illustrator, Adobe Photoshop etc). TIFF uses lossless compression and is often the highest quality format produced by digital cameras. However the file size of the resulting image is huge compared to JPEG and the quality difference is not always noticable to most people.

TIFF is suitable for image storage or manipulation where quality is important. You can edit a TIFF file with a software like GIMP and lose no quality when you save it as a TIFF. Excellent for storage of original image material or for print production. TIFF is not suitable for web browsers as the file size is huge and most browsers do not support displaying TIFF.

If you save a TIFF file you can use the suffix '.tif' or '.tiff' (or '.TIF and '.TIFF').

GIF

The **Graphics Interchange Format** (GIF) is owned by ... well theres a story in itself. It used to be owned by UNISYS and it is still in some countries but in others the patent has expired. In 2003 and 2004 the patent expired in the USA, United Kingdom, France, Germany and Italy, Japan and Canada - hence in these areas GIF is a patentless format and belongs to the public domain. The history of the ownership of GIF is interesting and worthwhile reading about if you have nothing better to do than read about histories of file formats!

GIF is a lossy format, however this is not quite true...GIF actually only uses lossy compression if the original image uses more than 256 colors. If you know how many colors your image contains and it is less than 256 then GIF will compress the file size a lot while keeping the image exactly the same. This makes it very effective for web graphics where small file sizes are important.

However, if you have more 'color rich' images (eg. most images from a digital camera) then converting them to GIF will reduce the file size but also the lossy compression will dramatically reduce the colors used and hence the quality will be reduced.

Hence GIF is ideal for creating simple images of text (for example) which might be used in a navigation bar on a web page. However your holiday snaps put online in GIF format will look pretty surreal.

GIF supports transparency which makes it nice for webpages where the background of an image needs to blend into the webpage background color. GIF transparency is supported in all browsers.

GIF also supports animation and there are several tools that enable you to make animations using GIF. These animations can be displayed in any browser.

When saving a file as GIF use the '.gif' or '.GIF' suffix.

JPEG

JPEG stands for **Joint Photographic Experts Group**. Huh? you might think..."I thought it was a file format not a committee"...well....the Joint Photographic Experts Group is the name of the committee that created the JPEG format. However there was a battle over who actually owns the format but it is now settled (phew). The JPEG committee believe the format should be used without enforceable license fees, so while the format is patented you can use it for free.

JPEG is the most commonly used format for images on the web. It is *usually* a lossy format but there are variations of JPEG that use lossless compression. It is likely that any form of JPEG you use reduces file size by lossy compression.

Where GIF will kill the quality of images with many colors, this is where JPEG excels. A photo from your digital camera will look good when compressed with JPEG while reducing the file size dramatically. Most digital cameras store images in JPEG for this reason.

The amount of data thrown away when saving to JPEG is determined by the software. Good softwares give you the choice of the compression level JPEG will use on your image. Generally speaking a compression ration of 85% will dramatically reduce file size while not creating any noticable quality deterioration. However the level of compression needed depends on your needs and your eye. Experiment with it.

JPEG is great for putting images with many colors online, and it is also great for reducing the storage space needed for your collection of digital images.

JPEG does not support animation or transparency.

You usually use **.jpg**, as a suffix for JPEG files although .jpeg, .jpe, .jfif and .jif are all used (or the capitalised equivalents : .JPG, JPEG etc)

41. IMAGE TOOLS

There are some very good Free Software image manipulation tools that you may find useful in preparing images for translation or translating tools.

GIMP

<http://www.gimp.org/>

GIMP (which stands for GNU Image Manipulation Program) or 'The GIMP' as it is sometimes called, is a very powerful image processing tool. You may already be familiar with software like Adobe Photoshop; Gimp is similar to Photoshop in its features and functionality and it can open Photoshop documents. Gimp can also export to Photoshop file formats so you can exchange images and working files with your colleagues and friends.

Gimp allows you to modify and adapt your images in many ways; you can resize images, crop them, or change the contrast and brightness amongst other things. You can also apply text to images, apply many different effects, or optimize images for print or for the web. Gimp can be used at home but it is also a tool for professional designers and image manipulators. You might find it has more features than you need if you just want to crop your holiday images, but you won't find it lacking if you are designing print or web material.

You can install Gimp on Mac OS X, Windows, or Linux. The Mac OS X install is a bit tricky and, it has to be said, a bit clunky to work with. This is shame as it deters many Mac users from trying this very fine tool. However installation on Windows is quite straightforward, and if you run Ubuntu (a type of Linux) then you are in luck - its already installed!

INKSCAPE

<http://www.inkscape.org/>

Inkscape is an open source drawing tool for creating and editing SVG graphics. More than just a text vector editor, Inkscape provides a WYSIWYG interface for manipulation of vector images, allowing the artist to express himself freely. While other free and proprietary software with similar capabilities exists, Inkscape provides an interface to directly manipulate the underlying SVG code. This allows one to be certain that the code complies to W3C standards. Since the beginning of its development, the Inkscape project has been very active, providing stability for the current software and the capacity for future growth.

Like other drawing programs, Inkscape offers creation of basic shapes (such as ellipses, rectangles, stars, polygons, and spirals) as well as the ability to transform and manipulate these basic shapes by rotation, stretching, and skewing.

Inkscape also allows users to manipulate objects precisely by adjusting node points and curves. Advanced artists find these functions indispensable in drawing software to freely create what they imagine.

A user can either manipulate the properties of objects individually and precisely through the XML editor or, in a more general and intuitive fashion, with input devices such as mice, pen tablets, and even touch screens.

In addition, Inkscape allows one to insert text and bitmaps (such as PNG—another W3C recommended bitmap image format) into an image, as well as perform some basic editing functions on them. If an artist requires further bitmap editing, he may use other tools (such as the GIMP) on images before or after importing them. If one does edit a linked bitmap in another program, Inkscape will reflect these changes once the SVG file is reloaded.

All of these characteristics make Inkscape a model drawing application, especially considering its flexibility and many other capabilities. Its strict compliance with the W3C SVG standards allows excellent portability of images to many applications and the varying platforms on which these applications run.

For more information, check out the FLOSS Manual on Inkscape:

<http://www.flossmanuals.net/inkscape>

42. WHAT IS AN SVG

SVG stands for Scalable Vector Graphic.

Images file formats like JPEG and GIF are made up of lines of individual pixels in a grid. This kind of image is known as a *raster* image. Vector graphics are a different type of image file altogether - they are not made up of pixels but consist of mathematical data and equations that describe the image.

Raster file formats, like JPEG, generally are created by either devices like digital cameras and scanners etc, or from software like GIMP or Photoshop. Vector graphics always originate from specialised vector graphic softwares such as Inkscape.

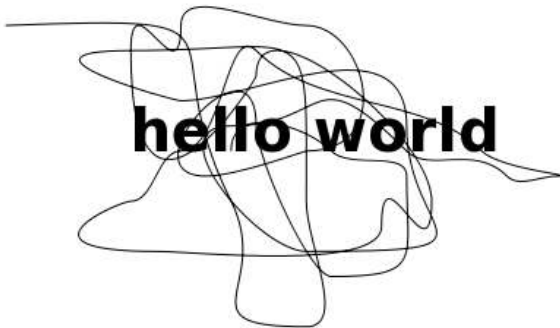
If you wish to create an image using software then you have a decision to make - do I create a raster or vector graphic? Generally speaking, you create raster images for most cases. However, there are some good reasons to consider using vector graphics. If you wish to be able to scale an image without losing quality then you would use vector graphics. Scaling raster images is usually hazardous and often results in a loss of quality and pixelation. This is because a raster image has a set number of pixels and scaling that image means adding or throwing away pixels. Whereas vector graphics scale without loss because you are simply changing the mathematics that describe the lines, dots, and curves of the image.

Just like there are many different types of raster file formats (JPEG, GIF, TIF, PNG etc), there are many varieties of vector graphics. SVG is one particular type of vector graphic file and it is the one primarily used by Inkscape. An SVG is usually identified by the name of the file - if it ends in '.svg' then it is a scalable vector graphic.

An SVG is also interesting because it is actually just a text file describing an image. This means that you can edit the file with a text editor or, more interestingly, it is possible to make simple programs that can edit SVG files. This holds very interesting possibilities for online translation of SVG files (useful only if an image contains text).

43. TRANSLATING SVG

SVG as a format allows translators easy access to text. For example, the following image was created as an SVG with text by Inkscape and exported as a png :

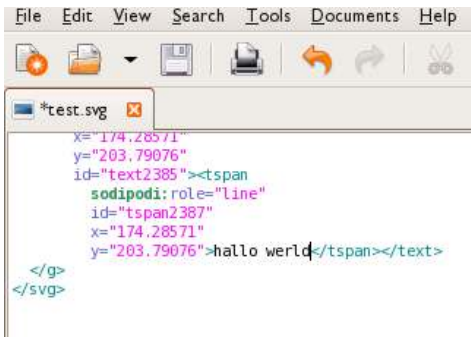


Because SVG is a text file you can edit an SVG image in a text editor. If I open the original SVG in a text editor I see something like this :

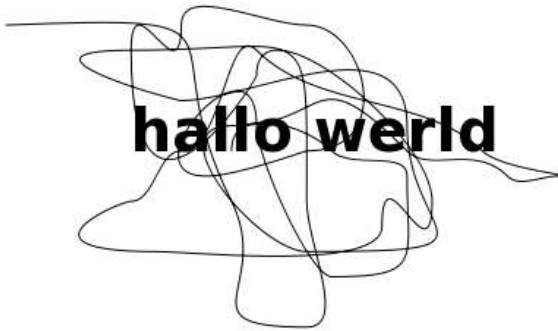
```
File Edit View Search Tools Documents Help
[Icons]
test.svg
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->
<svg
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:cc="http://creativecommons.org/ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:svg="http://www.w3.org/2000/svg"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
  xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
  width="744.09448819"
  height="1052.3622047"
  id="svg2"
  sodipodi:version="0.32"
  inkscape:version="0.46"
  sodipodi:docname="test.svg"
  inkscape:output_extension="org.inkscape.output.svg.inkscape">
  <defs
    id="defs4">
    <inkscape:perspective
      inkscape:persp3d-origin="372.24729,526.1811,1" />
  </defs>
  <g
    id="g1"
    transform="translate(0,0) rotate(0) scale(1,1)"/>
  </g>
</svg>
```

Now if I want to change the text "Hello World" and not change any of the other content in the image, I can scroll down until I see the text I want to edit :

```
File Edit View Search Tools Documents Help
[Icons]
*test.svg
  x="174.28571"
  y="203.79076"
  id="text2385"><tspan
    sodipodi:role="line"
    id="tspan2387"
    x="174.28571"
    y="203.79076">hello world</tspan></text>
</g>
</svg>
```



After saving, I can then open the file in Inkscape and I see the same background with the new translated text :



From Inkscape I can then export the file as a png or other image format. You can also edit the text with an SVG editor like Inkscape of course and translate the text, but with simple images, or for developing efficient workflows, a text editor might work just as well.

44. SCRIPTING SVG

Since SVG is a text file, it is very easy to use simple scripting techniques to edit SVGs. It is possible, for example, to write web-based programs so you can edit and translate SVG through your browser. This opens up very powerful possibilities for translating text content in an image.

Normally, translation of an image means opening the image in the appropriate image editor. With luck, you will have access to the source files of your image and hence the text exists as a separate 'layer' within the image. This makes translation of the text content reasonably easy.

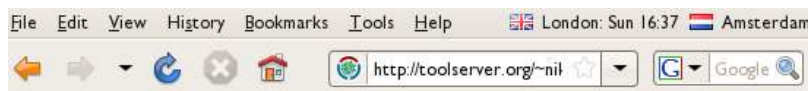
However this method does not enable easy translation of content online, and it also assumes you have the source files of the image AND the image editing software it was created with. If for example you have the source files for an Adobe Photoshop file but you don't have a license for Adobe Photoshop then you cannot translate the file.

Of course, if you don't have the source file then you are in a very difficult situation. To replace text in a bitmap file you need to delicately erase the text, rebuild the background, and then place the new text over the top...this can take a lot of time and can produce very ugly results depending on how complex your image is and how good you are at image manipulation.

SVG avoids all this because in a sense, every SVG is in itself an editable 'source file'. If you open an SVG in software capable of editing SVG, you can change any of the elements without effecting the other parts of the image.

SVG TRANSLATE

One example of the potential for SVG Scripting is "SVG Translate," an experiment in translating the text of an SVG online: <http://toolserver.org/~nikola/svgtranslate.php>



[SVGTranslate 1.05](#) [view source](#) [view include.php source](#)

SVG file URL:

NOTE: the SVG file URL must point to the file itself, not to its description page. For example, http://upload.wikimedia.org/wikipedia/commons/8/8a/Planetary_transit.svg is a valid URL. You may get this by going to the image description page, right clicking over the SVG and selecting "Copy Link Location" or similar in your browser's menu.

Questions, comments, suggestiong and bug reports are welcome at [my talk page](#).

Its really a proof of concept but it illustrates very well the power of scripting and SVG. Through the interface you can enter the URL of an SVG file. For example if I enter the URL for this SVG image :



and press 'Submit' I get this page :

File Edit View History Bookmarks Tools Help London: Sun 16:42 Amsterdam

http://toolserver.org/~nil Google

[SVGTranslate 1.05](#) [view source](#) [view include.php source](#)

Translating...

Original	Translation
hallo world	

Language: Afar

Submit Translation Reset

If I type the French translation into the given text field and press 'Submit Translation' the service replaces the text and offers the new SVG to download. If I open it in my SVG image editor I see this :



As you can see, SVG's text base makes it easily editable online and SVG continues to have massive potential for image translation and web-based manipulation.

INTELLECTUAL PROPERTY

45. Free Software?

46. LICENSING

47. Legal Considerations: MT and Copyright

45. FREE SOFTWARE?

Free Software (sometimes also referred to as **Free and Open Source Software**, **FLOSS**, **FOSS**, **Software Libre**, or **Open Source**) is software under a protective license guaranteeing that anyone can download, share, and -- significantly -- change it in any way they want, and redistribute the results. Practically speaking, you might never want to change the software yourself, or even have a resource person who can read the **source code** (the instructions written by programmers). But you can ask others to make changes for you.

The ability to change the software protects you in many ways:

- Free Software projects usually don't vanish, unlike non-free software from some companies that can no longer be licensed if the company goes bankrupt or decides to discontinue a specific product.
- The software can be used and customised free of charge.
- Nobody can suddenly take away features or change the terms under which you're allowed to use features.
- If an organisation wants a feature that the software doesn't provide, the organisation can just hire someone to create it. Of course, the organisation can also submit a feature request to the project team as with any software product.
- Similarly, anyone can fix a **bug** (error in the software) if he or she has the skills to do so. Because the source code is available, clients can also find bugs more easily.
- Members of the community have much more input into how the project develops, because they can understand the product by reading the code and can make changes. Furthermore, many people can try different implementations of features and the community can decide by vote or consensus which one to make official.

Nearly any software that qualifies as free also qualifies as Open Source, and vice versa. The main reason that two different terms exist is that "free software" emphasizes the freedom aspect (that you aren't under the control of the original programmers) whereas "open source software" emphasizes the convenience and potential for innovation provided by having the source code available.

When you install and use most Free Software applications you'll notice there's no annoying click-through **software license** imposing a thousand things that you can or cannot do with it. That's because free software doesn't limit your right to do with the software whatever you want. Free and open source software have licenses, but they're simpler than and quite different from **proprietary software** ('closed software') licenses.

46. LICENSING

Intellectual Property (IP) is an analogy for describing legal ownership of ideas. Common types of intellectual property include copyrights, trademarks, patents, industrial design rights and trade secrets. Copyrights are the legal area that apply most to Open Translation, as in the United States translations are deemed "derivative works," which may be violations of copyright. However, software patents for translation tools or machine translation algorithms may also be relevant, depending on a number of factors.

Outside the United States, whether or not translations are deemed "derivative works" is hotly debated and varies widely by jurisdiction. Note that part of the debate centers on the nature of translations themselves: are translations simply word-for-word substitutions from one language into another, or are they nuanced interpretations of the meaning of texts (as well as faithfulness where possible to specific words used) from one language to another? In the former case, an argument could be made that a translation is not a derivative work; in the latter case, it is virtually certain that the translation is a derivative work. This distinction matters because your perspective has an impact on the tools (including licenses) and goals of any open translation project. Note that many people have an expansive view of "translation" which includes format-shifting (such as rendering the content for disabled users) and other types of changes to the original work, which again suggests that translations are best understood as derivative works from an intellectual property perspective.

WHY LICENSE?

Licensing is controlling the rights for use and redistribution of your created work. *Licensing is very important.* In the United States, by default you own an "*all rights reserved*" copyright to any work that you create. In regards to translation, republication, and sharing over the web, it becomes very important to understand exactly what licensing rights you have, and the details of which rights you want to waive in order to further your project.

For example, if you waive certain aspects of your copyright (from an "*all rights reserved*" to a "*some rights reserved*" model), it can be beneficial to the quantity, quality, scale and success of your translation project.

Not only does copyright apply to translatable content, but also to the software tools that aid in the translation of that content. When we refer to 'Open Translation Tools', for example, we are generally referring to software tools licensed in a way that allows the free use, reuse, and alteration of the software.

COMMON TYPES OF LICENSES

Choosing the proper license for your software tool or translatable content is extremely important. Below are some common types of licenses and their distinctions.

Full Copyright

The default "all rights reserved."



Copyright was designed to allow for legal protection of authors, allowing them to hold exclusive rights to their work for a limited time. While copyright term was only designed to last 14 years in the United States, corporations have pushed forward legislation that has now expanded the copyright term to last the lifetime of the author plus 70 years from the original date of publication.

After this time, the work falls into the public domain--the free zone of sharing where everyone is free to use the work. Note that although all signatories of the Berne Convention on Literary and Artistic works have a limited (if overly long) duration for copyright, not all countries unambiguously deposit works out of copyright into the public domain.



Copyright protects content owners from having others steal their work and treat it as their own. However, many content creators can unintentionally prevent their work from being spread by not properly communicating (through licenses) which specific uses of their work they wish to allow (such as the right to save an image, document, or other file from the Internet.)

Permissive Licenses

"All wrongs reserved."

Permissive licenses allow the work to be re-purposed. From a legal perspective, permissive licenses often exist merely to disclaim warranty. This removes all liability by the copyright holder. *If it breaks, you can't say it's our fault.* By default a copyright has "all rights reserved." An easy way to think about permissive licenses is "all wrongs reserved." This kind of license says "we don't care about copyright." Some examples of software projects that use a permissive license are Apache Web Server and the Berkeley Software Distribution (BSD).



Permissive licenses can be good because they provide free and open source software, and free content, all of which are unencumbered by any significant restrictions on uses or users. Permissive licenses can be bad because modified versions of the software (for example), often improvements, do not have to be free like the original. For example, the proprietary, closed-source operating system Macintosh OS X is based off the Berkeley Software Distribution. Apple Macintosh is basically "free-riding" on the backs of the diligent coders who create BSD.

Some important free tools released under this license are Optical Character Recognition (OCR) tools such as Tesseract and OCRopus, both released under the Apache license. Google currently funds and uses both of these tools in its large book scanning project. Note that the decision steps regarding the use of permissive licenses often vary considerably depending on the nature of the content--the differences among software and other types of content are particularly important.

Copyleft Licenses

"Sharing is caring."

The "problem" of future versions not being free is solved by copyleft licenses. 'Copyleft' as an idea is a largely ethical, philosophical, and political movement that seeks to free ideas from the constraints of intellectual property law. According to the proponents of copyleft, duplication via communication is part of the very essence of what it means to have an idea, that is to say, what it means to share an idea. As they say, "sharing is the nature of creation."



The earliest example of a copyleft license is the GPL. Written in 1989 by a computer scientist at MIT named Richard Stallman, the license was designed to ensure the future success of Stallman's project: GNU. GNU, or *GNU's Not Unix* (a snarky recursive acronym I might add), is a massive collaboration project in free, community-developed, community-maintained software.

The GNU project has grown to enormity in its success, with the GPL quickly being adopted by free software projects across the world. Today, most serious and self-respecting open source projects publish their code under the GPL, which simply stands for GNU Public License. However, the freest license on the Internet today is not in fact the GPL, but the Affero GPL or AGPL. This license is only used for a few projects, such as Laconica.

Since FLOSS Manuals is designed as a documentation service for Free (as in Libre) and Open Source Software, our text is published under the GPL so that it can be redistributed with the free software that it is a reference for. This text, since it covers some open source tools, is released under the GPL as well.

Creative Commons Licenses



"Saving the world from failed sharing."

Creative Commons (CC) licenses are several copyright licenses first published on December 16, 2002 by the Creative Commons organization, a U.S. non-profit corporation founded in 2001 by a Stanford law professor named Lawrence Lessig. "Larry" has written several books on US copyright law--his latest is called *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. The goal of the Creative Commons organization is to enable effective sharing of copyrighted content. Larry is also featured in a fantastic documentary called *RiP: a remix manifesto*. If you're too young to drink in the states, watch the movie. If you're too old to get smashed like a kid in the states, read the book. If you are in neither category do both. Then you will understand.

Attributes of CC Licenses

The Creative Commons organization sought to create a common language for licenses, so that they were readable and immediately understandable by the everyday man. They created a spectrum of free licenses, with varying freedoms controlled by various attributes. *These attributes can be mixed and matched to create a custom license*. Not all combinations are possible since some attributes are mutually exclusive. Here are the various licenses composed of the four basic attributes:

Attribution (CC-BY)

Attribution (BY) comes by default with all CC licenses, with the only exception being the CC0 license. This attribute is to say *"give credit where credit is due."* Copying is only permitted when the author or authors of the original work are properly attributed. You may choose to have attribution be the only stipulation (CC BY license), in which case the work can then be repurposed for anything. *This is essentially a permissive license.*



Attribution Share-Alike (CC-BY-SA)

Share-Alike (SA) means that any derivatives of the original work *must be shared under the same exact license as the original*. This option allows for the freedom to 'remix,' but backed by a principle of reciprocity. I share with you, you share with everyone else. The CC BY-SA license also requires that you properly attribute the original author, like all CC licenses. *This is a copyleft license.*



Attribution No-Derivatives (CC-BY-ND)

No-Derivatives (ND), like the rest of these terms, is fairly straightforward. If present, you may not make derivative works of the original work. The right to 'remix' is reserved by the original author, who may wish that his work is not mutilated or misrepresented. While this term can be used in a free license as it allows for copying, *licenses using the ND term are less free than permissive or a copyleft licenses*. Licenses using the ND term are less free is because they make for works that are sterile and not generative. I use the terms "sterile" and "generative" in the Jonathan Zittrain sense.



Because no derivatives are allowed, the ND and SA terms cannot be combined. Note that the lack of permission to make derivative works means that the work may not be translated (under most circumstances). As such, Open Translation projects are not likely to either use or recommend the use of the ND term.

Attribution Non-Commercial (CC-BY-NC-[ND/SA])

The Non-Commercial (NC) attribute is a stipulation that you may reuse the work provided that it is only used for non-commercial purposes. The author reserves the right to monetize and commercialize the work. This means the work cannot be sold by any person or corporate entity, not even a non-profit corporation. However, the work may be shared freely and, depending on the actual license chosen, adapted (including translated) or modified without permission. If the share-alike (SA) term is also used, then those modifications must be relicensed and made available using exactly the same license. If the no derivatives (ND) term is used, then the original work can only be copied and distributed, not adapted, and then only non-commercially. *As with the ND term, licenses with this attribute are less free than permissive or copyleft licenses.*



It is worth noting that the interpretation of the non-commercial restriction is not clear-cut. In addition, most proponents of copyleft licensing for software (generally with the GPL) find the commercial constraint to be highly discriminatory to legitimate open business models and believe that the NC term should be avoided at all costs.

The CC0 Waiver

"No rights reserved."

The CC0 waiver is *not* a license, but is rather a tool that allows a copyright holder, to the extent possible, to release all restrictions on a copyrighted work worldwide. CC0 was created (at least in part) in response to new database rights ("moral rights") in the European Union. Because copyright law is different for databases in the European Union, the Creative Commons organization combined a public domain dedication with a waiver that released any and all ownership rights, including those on data. The CC0 waiver is the closest thing to a public domain dedication, *as you waive any possible rights.*

THE PUBLIC DOMAIN

The public domain is the free realm of sharing where the original author, if known, contains no rights at all over his work. Some freedom-lovers who seek no recognition for their work simply release it into the public domain with a tool called a *public domain dedication*. Additionally, when the copyright term expires for a particular work, that work enters the public domain (at least in the United States). Since work in the public domain lacks any copyright protection, it may be leveraged by corporations hoping to profit from the work, as well as anyone else. This is common for reprinting of out-of-copyright texts like Alice in Wonderland, Shakespeare, and other classics. Once a work is in the public domain, the original work can never be copyrighted by anyone, regardless of any copyright claims on new renditions.

47. LEGAL CONSIDERATIONS: MT AND COPYRIGHT

American copyright law considers a translation a derivative work. As such translators must obtain permission from the copyright or derivative right holder of the source language text. With regard to online translation, we expect that as Machine Translation (MT) and Hybrid Distributed Translation (HDT- strategies combining human and machine translation) come of age significant changes will need to be made to the legal framework to accommodate these technologies.

In his excellent research paper, *Rebuilding Babel: Copyright and the Future of Machine Translation Online* (http://papers.ssrn.com/sol3/papers.cfm?abstract_id=940041), the legal scholar Erik Ketzan posits that Machine Translation (MT) will “create massive copyright infringement on an unprecedented global scale.” Ketzan argues that “the law needs to pave the way for companies to develop online translators [MT] and forestall a chilling effect on innovation that may result from legal uncertainty; software companies may not pursue online translators because of the threat of litigation. Online MT needs to be protected because it is socially, politically, and commercially beneficial. Technology may have put man on the moon, but machine translation has the potential to take us farther, across the gulf of comprehension that lies between people from different places.” Ketzan argues the need, “for protection through the creation of effective licenses and statutory clarification of online MT’s noninfringing nature.”

We believe there is an unmet need for research and legal advocacy addressing the status of both Machine Translation and social translation of webpages and other digital content.

First, though, courtesy of Ketzan, let us take a quick stroll through the history of translation copyright in the west.

Among the notable unfortunate moments in the history of translation copyright, William Tyndale (patron saint of translators) was given neither a take down notice nor a cease and desist but was instead burned at the stake for his unauthorized English translation of the Bible in 1536. [THE RADICAL REFORMATION xxvii (Michael G. Baylor et al. eds., Cambridge University Press 1991); David Daniel, *William Tyndale: A Biography* (Yale University Press 1994)] England's first copyright law, The 1710 Statute of Anne, did offer publishing protection for a fixed number of years, but did not protect authors with regard to translation rights. Two 18th Century court cases did address the issue, concluding: "a translation might not be the same with the reprinting the original, on account that the translator has bestowed his care and pains upon it." [Burnett v. Chetwood, 2 Mer 441, 35 Eng Rep 1008-9 (Ch 1720) via Ketzan] And, "Certain bona fide imitations, translations, and abridgments are different [from copies]; and in respect of the property, may be considered new works: but colourable and fraudulent variations will not do." [Millar v. Taylor, 4 Burr 2303; 98 Eng Rep 201 (1769) via Ketzan] In fact it was not until 1911 that English law granted a work's author the right to control translations.

Ketzan continues, "American law did not recognize unauthorized translations as copyright violations until the late nineteenth century. [Naomi Abe Voegtli, *Rethinking Derivative Rights*, 63 Brook. L. Rev. 1213, 1233 (1997)] In *Stowe v. Thomas*, a Pennsylvania District Court held that an unauthorized German translation of *Uncle Tom's Cabin* (German being commonly spoken in Pennsylvania) did not constitute a "copy" under copyright law. [Stowe v. Thomas, 23 F. Cas. 201, 207 (C.C.E.D. Pa. 1853)] ("A "copy" of a book must, therefore be a transcript of the language in which the conceptions of the author are clothed; of something printed and embodied in a tangible shape. The same conceptions clothed in another language cannot constitute the same composition, nor can it be called a transcript or "copy" of the same "book." ") Congress explicitly reversed this holding in the 1870 Copyright Act, which recognized a form of derivative right in translations by providing that "authors may reserve the right to dramatize or to translate their own works." [Act of July 8, 1870, ch. 230 § 86, 41st Cong., 2d Sess., 16 Stat. 198] The 1909 Act maintained and expanded this translation derivative right, granting authors the right to 'translate the copyrighted work into other language or dialects.'" [http://papers.ssrn.com/sol3/papers.cfm?abstract_id=940041]

What emerges from Ketzan's analysis is a history of translation copyright in which absence of rights followed from the linguistically defined boundaries of the publishing industry of the day. There was no incentive to lobby for translation rights, because the publishing houses did not work across markets. This offers clear analogy to social online translation. The structural fact that content producers (with notable exceptions) are generally not targeting and have not good way of monetizing a single piece of content to a global audience, works to the favor of online translators. Linking a Japanese speaker who has read a lengthy translation of an article from Huffington Post back to that English language site, is meaningless to the user and to Huffington post.

Back to a consideration of legal issues related to MT.

The copyright implications of MT have not been addressed by a Federal court in the United States. As mentioned above, the exclusive right to authorize derivative works belongs to the owner of the copyright. Ketzan: "It is beyond question that translations constitute derivative works, which are actionable if not authorized by the copyright holder of the original. [7 U.S.C. § 101 (definition of "derivative work" includes "translation"). See also 1-2 NIMMER ON COPYRIGHT, supra note 122, at § 2.04, §3.01.] But is MT output actually a "translation," and therefore a derivative work, under the Copyright Act? Just as the copyright laws do not expressly require "human" authorship. [Urantia Found. v. Maaherra, 114 F.3d 955, 958 (9th Cir. 1997) (addressing the bizarre question of whether a book purportedly authored by celestial beings may be copyrighted; "The copyright laws, of course, do not expressly require "human" authorship")]. Title 17 does not explicitly require translation, or any other derivative work, to be performed by a human. Sound recordings and art reproductions, like MT, can be created at the touch of a single button and create derivative works under § 101. While an argument can be made that, theoretically, MT is not "translation," [LE TON BEAU DE MAROT, supra note 40, at 515-518] a plain language reading suggests that machine translation performs what it says: translation. As such, machine translation of a text creates derivative work under the Copyright Act and [an MT service provider] may be liable for copyright infringement if that translation is unauthorized."

The notion of 'implied license' is a well established concept that a content owner who publishes to the web understands and agrees to enable web crawlers to index their content and web browsers to change their CSS and and fonting. In a Nevada court ruling [Field v. Google, Inc., 412 F. Supp. 2d at 1115-16] it was ruled that the burden is on the website owner to opt out of Google indexing. EG, the web-page owner has the ability to opt out of Google indexing therefore cannot argue that rights are infringed by Google. Currently, Google translation services note to the end user that they should use: `<meta name="google" value="nottranslate">`

The notion of 'fair use' cannot be invoked for translations that constitute full web-page translations due to the fact that one of the considerations for meeting the 'fair use' criterion is "the amount and substantiality of the portion used in relation to the copyrighted work as a whole." In the case of a full page machine generated translation, MT fails to meet 'fair use.'

Ketzan asserts that MT service providers could find protection under the Digital Millenium Copyright Act (DCMA), which offers broad immunity to Internet Service Providers (ISP) for content that might be presented over website they serve. The question is whether an MT company could be defined as a 'service provider'. The DMCA defines a service provider as "a provider of online services or network access." This is perhaps the strongest legal defense for companies that provide provide MT services onto the web.

Social Translation projects such as Global Voices and Meedan.net preform social translation of content on the web (Meedan.net also provides Machine Translation services similar to Google). The projects assert their translation work is a non-commercial public service, they provide links to original sources, and they encourage any content producers who do not want translation services to notify them.

If there was a global equation to describe the velocity of innovation, collaboration, knowledge creation, and knowledge sharing--a sort of *global understanding index*--it would be limited by the scope and rate of the transfer of knowledge and information across language communities. Limiting this flow of information limits our ability to 'compete globally' (to borrow a phrase with irony) against our baser tendencies toward xenophobia, ignorance, and narrow thinking. We need a worldwide agreement that the worldwide web is, indeed, worldwide, and that in publishing a piece of content to the web, we should embrace the idea that people from outside our language have a right to put on a pair of glasses (MT algorithms/HDT services) that might allow them to attempt to decipher our words and ideas. Efforts to restrict translation rights may similarly limit our ability to successfully navigate our inevitably global future.

OPEN SOURCE TOOLS

48. Anaphraseus

49. Apertium

- 50. Gaupol
- 51. Glossmaster
- 52. Moses
- 53. Okapi Framework
- 54. Somebody Should Set The Title For This Chapter!
- 55. OmegaT
- 56. OpenOffice.org
- 57. Pootle
- 58. Translate Toolkit
- 59. Virtaal
- 60. Worldwide Lexicon (WWL)

48. ANAPHRASEUS



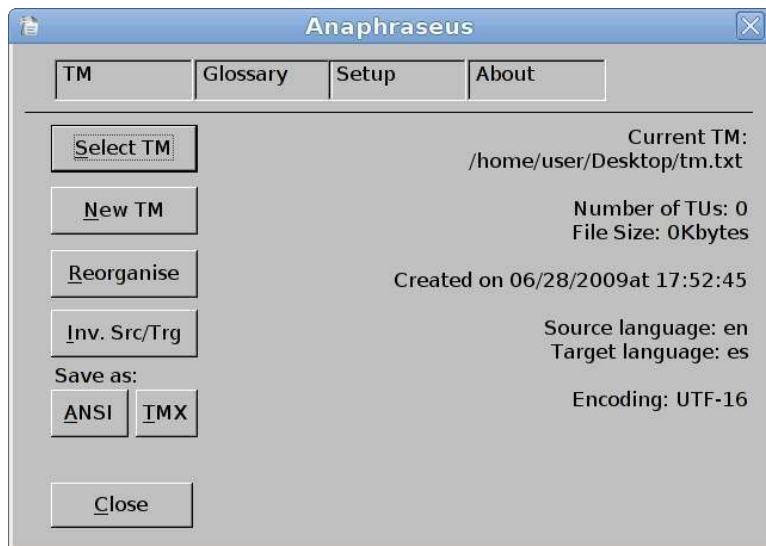
Anaphraseus is a computer assisted translation tool consisting of a set of macros for OpenOffice Writer, basing on the same principle of WordFast and Trados. It is available both as an OpenOffice.org extension and as a standalone document with macros, and it can be used on Windows, GNU/Linux, FreeBSD and MacOS X.

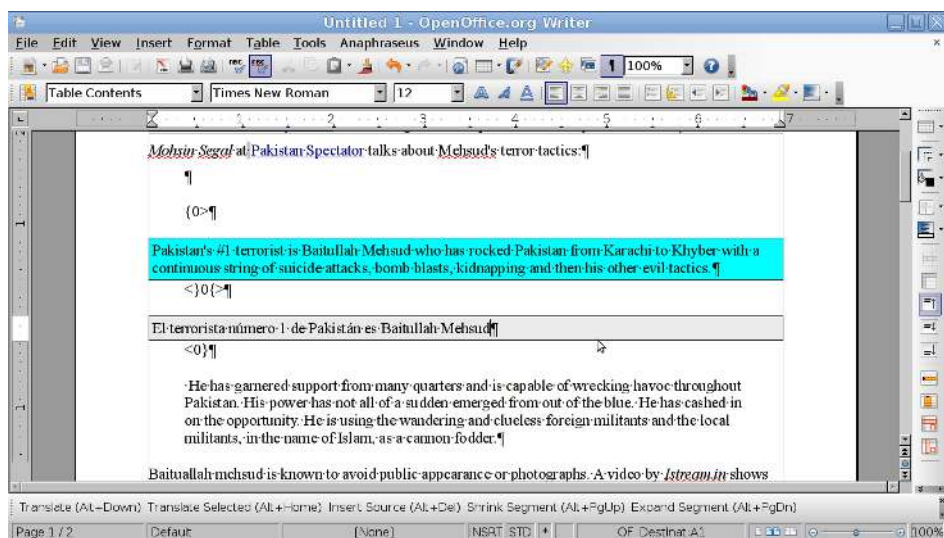
WHAT IS IT USED FOR?

It can be used for translating files in any of the formats supported by OpenOffice Writer (txt, rtf, odt, etc.) with the aid of translation memories and glossaries. It can also be used to 'clean' files in Trados or WordFast 'unclean' format.

Its main features include:

- Text segmentation
- Terminology recognition
- Plain-text TM (Unicode UTF-16)
- Fuzzy search in Translation Memory
- Unicode UTF-16 TMX export/import
- User glossary





RESOURCES

- Project website: <http://anaphraseus.sourceforge.net/>
- Anaphraseus feedback mailing list: <https://lists.sourceforge.net/lists/listinfo/anaphraseus-feedback>
- Anaphraseus forum: http://sourceforge.net/forum/?group_id=207409

Anaphraseus is free software licensed under the GPL.

49. APERTIUM



Apertium is a software tool for translating one language into a closely-related other through the use of a computer program rather than human interpretation. Apertium uses a language-independent translation work flow, has tools to manage the linguistic data surrounding the two languages (a "language pair") and already contains linguistic data for many language pairs which are encoded in an XML-based format.

WHAT'S IT USED FOR?

Apertium is used mainly for translating closely-related language pairs. Originally produced as part of the OpenTran project funded by the Spanish government, Apertium's strength is its strong translation support for languages closely related to Spanish. For example, Spanish-Portuguese or Spanish-Catalan. Its focus on other languages has been limited due to its history. However, because it is open source and has such a strong community, users have been able to push Apertium into more dissimilar language pairs with the hope that the platform will be able to go much further than languages similar to Spanish.

With Apertium, users are able to create machine translation systems for different languages by cataloging the linguistic data in a specific XML format that Apertium can then interpret. That interpretation is through a process somewhat like an assembly line with each piece of language going through the following steps:

1. *De-Formatting*: stripping the formatting (e.g. HTML or RDF) from the text
2. *Morphological Analysis*: assigning grammatical designations for the text
3. *Part-of-Speech Disambiguation*: determining parts-of-speech for uncertain terms
4. *Shallow Structural Transfer*: processing phrases for differences in linguistic grammar
5. *Lexical Transfer*: assigning the right linguistic term from the other language
6. *Morphological Generation*: assigning grammatical designations for the new text
7. *Re-Formatting*: restoring the original formatting to the new text

RESOURCES

There are many resources for the Apertium platform that can help out the new user:

- Apertium Home Page: <http://www.apertium.org/>
- Apertium Wiki: http://wiki.apertium.org/wiki/Main_Page
- Apertium Mailing List: <https://lists.sourceforge.net/lists/listinfo/apertium-stuff>
- Apertium IRC (Chat): <http://xixona.dlsi.ua.es/cgi-bin/cgiirc/irc.cgi>
- Apertium News RSS: http://sourceforge.net/export/rss2_projnews.php?group_id=143781&rss_fulltext=1

Apertium is Free Software licensed under the GPL.

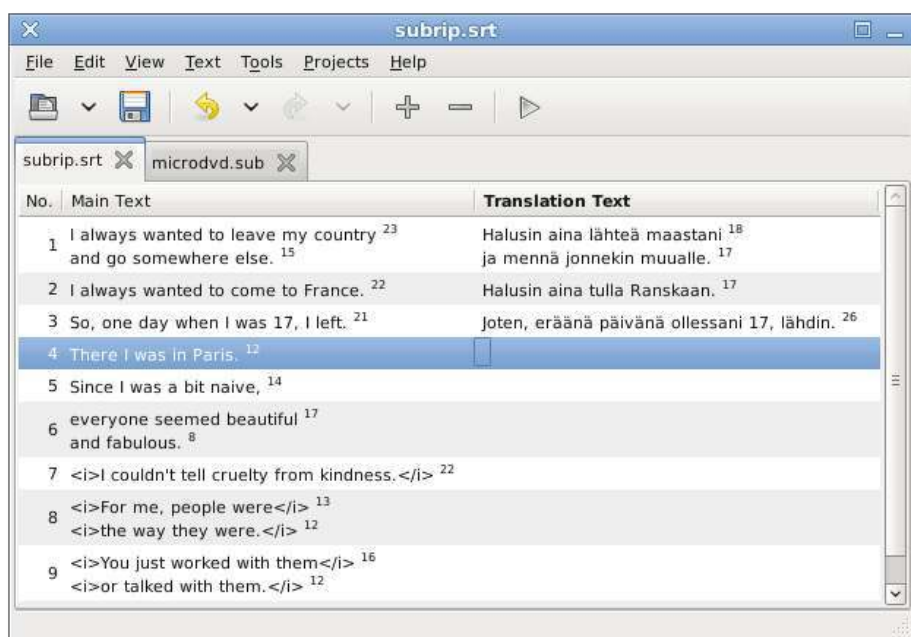
50. GAUPOL



Gaupol is an open source software tool for translating subtitles within video. Gaupol works with open source systems such as Arch Linux, Debian, Fedora, FreeBSD, Frugalware Linux, Gentoo Linux, NetBSD, openSUSE and Ubuntu, but it will also work with Windows.

What is it used for?

Created to translate previously-created subtitles, Gaupol is not made primarily for initial subtitle creation or editing the original subtitles in the video. Like many programs used for subtitles, Gaupol gives you parallel columns to translate line by line.



As a translation tool, Gaupol is simple to use for text-based subtitles with many small but useful features like the ability to find and replace text, framerate conversion and previews of the edits in an external video player. Gaupol is designed so that users can easily translate a group of subtitles at the same time and assign each to a specific time in the video.

Gaupol works in the following subtitle formats:

- Fully supported formats: MicroDVD, MPL2, MPsub, SubRip, Sub Viewer 2.0 and TMLayer
- Partially supported formats: Sub Station Alpha and Advanced Sub Station Alpha

Resources:

- Gaupol Home Page: <http://home.gna.org/gaupol/>
- Gaupol Development: <http://home.gna.org/gaupol/development.html>

Gaupol is free software licensed under the GPL.

51. GLOSSMASTER

Glossmaster is a glossary tool that provides a list of 2500 information technology terms in English with definitions. Many of the terms provided are from software programs written in English along with any other helpful information.

WHAT IS IT USED FOR?

With Glossmaster's core listing of terms and definitions, language teams can then easily translate the terms and definitions into local languages. For example, Glossmaster was originally developed for the African Network for Localization (ANLoc) and many such teams used Glossmaster in Africa to translate information technology terms and concepts into culture-understandable ways so that more people can be introduced to technological resources and concepts. Glossmaster's focus on African language is its main asset as a translation tool, however, the capacity for expansion into other non-African languages is there.

TRANSLATE STATISTICS EXPORT USER SETTINGS

Quick Search

☒ Manual ☐ All empty ☐ All fuzzy

Search string

Enter * to search for all strings

Syntactic group

All

Number of posts to display

25

Status

All
Empty
Fuzzy
Completed

Type of entry

Core

Search

Number of hits: 2500

1. 3-D

Definition

Comment

Edit

2. abbreviation

Edit

3. abort

Edit

4. aborted

Edit

5. aborting

Edit

6. about

Edit

7. above

Edit

8. absolute path

Edit

RESOURCES

- Glossmaster Home Page: <http://www.it46.se/glossmaster/>

52. MOSES



Moses is a translation tool that allows users to train the computer program to translate between a specific "language pair" (the two languages being translated) by inputting a collection of translated texts into a software program. The program then analyzes the content of the documents and is able to translate new documents based on statistical reasoning where it inputs the most likely translation of the text based on the usage from the translated source documents.

WHAT IS IT USED FOR?

As a statistical translation tool, Moses is used for translating content when there is a large corpus (a collection of sentences and their direct translations into the target language) that can be pulled in the system. However, even if the corpus of the languages being translated is small, Moses is still able to provide statistical translation as long as there is some data to work with.

Moses features include:

- Beam-search: An effective kind of search algorithm
- Phrase-based functionality: Allows translation of text segments and chunks
- Factored recognition: Recognizes different "factors" of language (e.g. part-of-speech, morphology)

RESOURCES

For more information on Moses, check out these resources:

- Moses Home Page: <http://www.statmt.org/moses/index.php?n=Main.HomePage>
- Moses Training: <http://www.statmt.org/moses/?n=FactoredTraining.HomePage>
- Moses Manual PDF: <http://www.statmt.org/moses/manual/manual.pdf >

Moses is free software licensed under the LGPL.

53. OKAPI FRAMEWORK



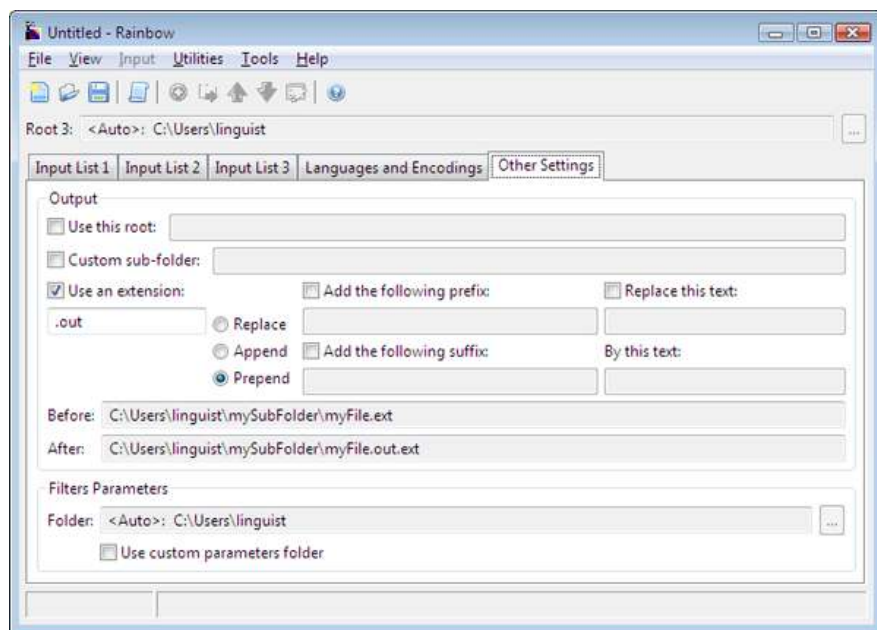
The Okapi framework is a set of cross-platform tools and components to help in translation and localization tasks. Those applications are built on top of Java libraries that you can also use directly to develop your own programs and scripts.

WHAT IT IS USED FOR?

The framework is useful to develop workflows to process files before and after translations, or to perform different tasks on translation-related data at any point.

For example, you can use Rainbow, one of the tools built on top of the libraries, to prepare for translation documents in many different formats, and to post-process them after translation. It supports the creation of different packages you can translate using tools such as OmegaT, Virtaal, and even commercial tools.

Other utilities include: encodings conversion, source-target text alignment, pseudo-translation, translation comparison, RTF conversion, search and replace, and more. The framework allows also to create new utilities easily using a pipeline mechanism.



The framework provides a collection of filters. They all have a common API and can be used to perform different types of actions on the translatable content of many different file formats. The filters include support for formats such as HTML, XLIFF, TMX, PO, XML (supports ITS), OpenOffice.org (ODT, ODS, ODP, etc.), MS Office 2007 (DOCX, XSLX, PPTX, etc.), Properties, CSV, and more. Additional formats can be supported by defining custom configurations, for example using regular-expressions based parameters.

RESOURCES

- Main Web site: <http://okapi.opentag.com/>
- GoogleCode project: <http://code.google.com/p/okapi/>
- Mailing list and users group: <http://tech.groups.yahoo.com/group/okapitools/>
- Developer's guide: <http://okapi.opentag.com/devguide>

54. SOMEBODY SHOULD SET THE TITLE FOR THIS CHAPTER!



OmegaT+, a computer aided/assisted translation (CAT)/machine aided human translation(MAHT) tool, has many of the good features that users expect in a translation tool (translation memory, full and partial matches, glossary function, search engine, support for various document types, translation projects) presented in a straight forward manner that is simple and easy to use.

APPLICATION AREA

OmegaT+ assists a human translator in the translation of documents by leveraging legacy translations contained in translation memories (TMX). A variety of document formats for translation are supported, including: OpenDocument (OpenOffice/ODF), HTML/ XHTML, plain text, Java properties, Portable Object, DocBook and others. During translation the translation memories are searched to provide matches that can be used to speed up the process of translation. The provided matches, from legacy translations, may be used and edited into a new translation that upon completion will be saved into translation memory again to provide further matches in future use. This cycle of use and reuse is continued until all segments (subparts of documents, usually sentences) in a document and ultimately a project (a number of documents grouped together for convenience or business purposes) are completed. The final result will be a fully translated translation memory that can be exported to TMX and used to regenerate the original documents with the translated text in its place.

OmegaT+ features:

- Exact and approximate Matches
- Automatic propagation of matches from project translation memory
- Supports multi-document projects
- Supports multiple external/legacy translation memories
- Supports right-to-left languages
- Nice interface with docking desktop, different styles, custom fonts, drag'n drop, and so on
- Supports translation memory (TMX) standard - prevents vendor lock-in
- Localized into a number of languages (can be switch between them from menu) and is simple to extend to new languages
- Cross-platform application. Runs on any Java supported operating system (Linux, Mac OS X, Windows, etc.)

OmegaT+ Home: <http://omegatplus.sourceforge.net>

OmegaT+ Project: <http://sourceforge.net/projects/omegatplus>

55. OMEGAT



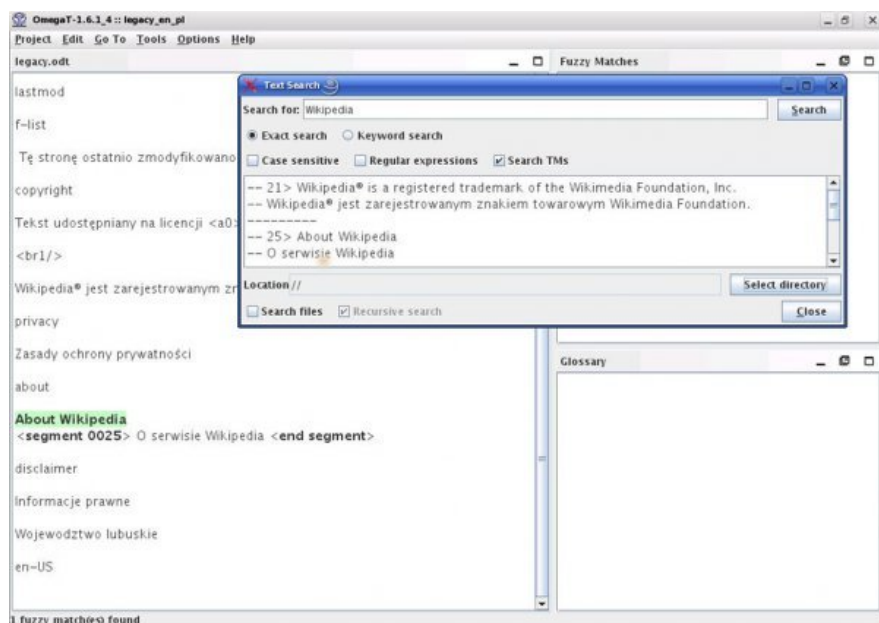
OmegaT is a free translation memory software tool that assists with translation on a computer. A tool intended for professional translators, OmegaT assists the user by managing source documents, translation memories and glossaries in translation projects.

WHAT IS IT USED FOR?

Contrary to "machine translation" tools which translate content *for* you, OmegaT assists the human translator with translation resources. It is used for translating content in a wide variety of formats (including OpenOffice documents, HTML and others) and has many features that help when translating materials. When translating a document with OmegaT, it uses its translation memories to suggest to the translator what it thinks segments of text mean. Suggestions may or may not be taken, but when the translator *does* decide what she thinks the text means, OmegaT stores her translation in the translation memory which it will then suggest as a possibility later if that same segment is come across. In this way, the translator retains ultimate control over the content of the translation but is given an extremely helpful tool for assistance.

OmegaT features include:

- Fuzzy Matching
- Match Propagation
- Simultaneous processing of multiple-file projects
- Simultaneous use of multiple translation memories
- Support for right-to-left languages
- Compatible with other translation memory applications (TMX)



RESOURCES

- Project page: <http://www.omegat.org/>
- Documentation and tutorial: <http://www.omegat.org/en/documentation.html>
- Mailing list and user group: http://www.omegat.org/en/contact_support.html
- The translated manuals: <http://tech.groups.yahoo.com/group/OmegaT/files/2-%20Documentation/> (yahoo login required)

OmegaT is free software licensed under the GPL.

56. OPENOFFICE.ORG

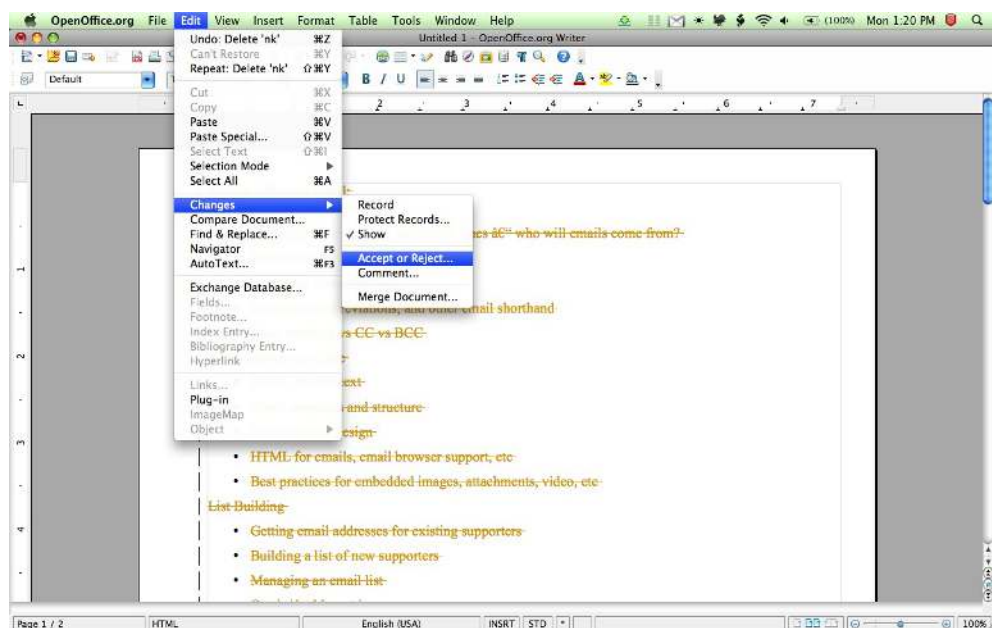


Open Office is a productivity suite that includes a word processor, a spreadsheet program, presentation software as well as an illustrating program. Open Office has many features that make it a standard tool among open translators.

What is it Used for?

Open Office is used primarily for administrative work with regard to word processing, spreadsheet calculations and presentations, however, Open Office can also be used as a translation tool. Many translators use text editors as a base tool and Open Office offers significant features for assisting translation work.

One such feature is the ability to record changes within a document. In translation, this is useful for tracking changes throughout the different versions of the document or text. Open Office's "Record" feature that includes comments and the ability to accept and reject changes as well as the "Compare Document" feature are useful additions to the translator's workflow when translating within a word processor.



Another feature useful for translation is the ability to open and save files in a variety of formats. For example, if a colleague sends a version of a translated document in Microsoft Word format, OpenOffice is able to open and save the format so that conversion is less of an issue.

Resources:

- The OpenOffice.org Home Page: <http://www.openoffice.org>
- The OpenOffice.org Wiki: http://wiki.services.openoffice.org/wiki/Main_Page
- Open Office Extensions: <http://extensions.services.openoffice.org/ >

57. POOTLE



Pootle is a web-based translation management tool that helps groups collaborate around language translation. Although web-based, the software is downloadable so that users can set up their own Pootle server on an intranet where people can contribute and collaborate. Some of these collaborative features include work assignment, statistical information, the ability to suggest translations and a database of previously-translated segments of language called "translation memory." Pootle is built on the Translate Toolkit, a free software toolkit for translating content.

WHAT IS IT USED FOR?

Pootle is mainly used for group collaboration around translating content. Translating the user interface of computer programs from one language to another was Pootle's intended use (as opposed to translating documents or subtitles). However, Pootle's project management features allow it to be used for a variety of translation projects that involve multiple people.

Pootle has been used by several Open Source projects such as Mozilla Firefox, OpenOffice.org and One Laptop Per Child.

RESOURCES

For more information about Pootle, check out these links:

- Pootle Home Page: <http://translate.sourceforge.net/wiki/pootle/index>
- Demo Pootle Server: <http://pootle.locamotion.org/>

Pootle is Free Software software released under the GPL.

58. TRANSLATE TOOLKIT

The Translate Toolkit is a collection of useful tools for localisation, and an API for programmers of localisation tools.

WHAT IS IT USED FOR?

The Toolkit is useful in translation workflows for simplifying formats. It can convert between various different translation formats (such as Gettext PO formats, XLIFF, OpenOffice.org, and Mozilla formats) giving you the option of working in one localization format for your entire translation work. The project itself seeks a better standardization of translation formats. Because of its formatting conversion abilities, the Translate Toolkit is useful, for example, for updating old translation files for newer templates.

The toolkit also includes tools to help check, validate, merge and extract messages from localizations. For instance, pulling a terminology list from a localization is easy through the Translate Toolkit interface as is finding differences in translations throughout one translation workflow.

RESOURCES

- Project page: <http://translate.sourceforge.net/wiki/toolkit/index>
- Localisation guide: <http://translate.sourceforge.net/wiki/guide/start>
- Translate Toolkit Use Cases: http://translate.sourceforge.net/wiki/toolkit/index#use_cases

The Translate Toolkit is free software licensed under the GPL.

59. VIRTAAAL



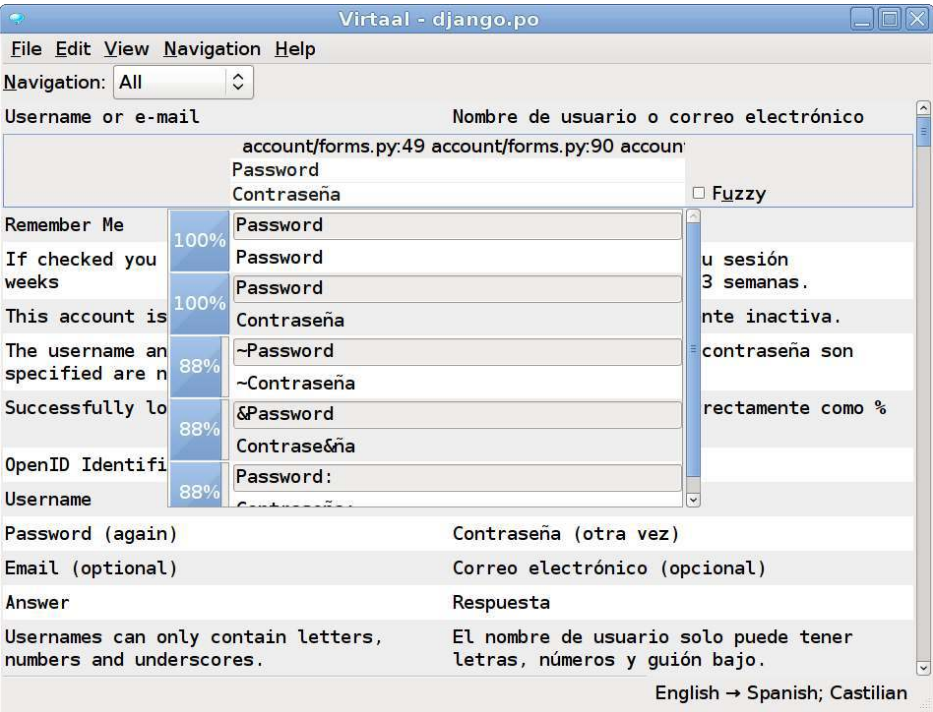
Virtaal is a desktop translation tool. It aims to remain easy to use for beginners while offering experienced translators a powerful tool.

What is it Used For?

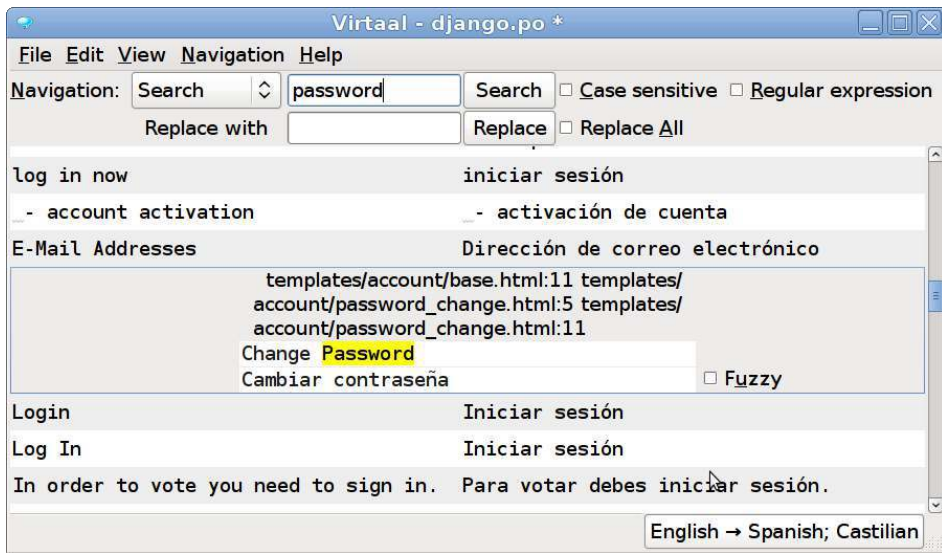
Virtaal is used as an assistance tool for translating content. Virtaal was intended for software localization however its interface is suited for other types of translation as well. It does not translate content for you, but rather incorporates "translation memories" and easy to access and use features to increase the speed and ease of a translation.

Virtaal can access many forms of translation memory through a system of plugins. By enabling various plugins a user can get translation memory suggestions from Google Translate, Open-tran.eu, the current file and even a shared translation memory server.

The translation memory is displayed along with the editing Window. Results from Machine Translation show no match percent and should be reviewed by a translator. Those from a Translation Memory will have a math percentage to measure probability.



Virtaal also allows searching within your translations. You can search within the source text or your translations. Modes allow a Virtaal user to change their editing strategy. You can set Virtaal to move to untranslated segments or to move through all segments.



Virtaal has support for the following localization file formats:

- Gettext (.po, .mo)
- XLIFF (.xlf)
- TMX
- TBX
- WordFast TM (.txt)
- Qt Linguist (.ts)
- Qt Phrase Book (.qph)
- Gettext (.po and .mo)

Other Features include

- Highlighting of XML and escape characters
- Displays comments from programmers and previous translators
- Displays context (like msgctxt in PO)
- Auto-correction of common mistakes
- Auto-completion of long words
- Reuse of existing translations

Resources:

- Virtaal Home Page: <http://translate.sourceforge.net/wiki/virtaal/index>
- Guide to Using Virtaal: http://translate.sourceforge.net/wiki/virtaal/using_virtaal

Virtaal is free software licensed under the GPL.

60. WORLDWIDE LEXICON (WWL)

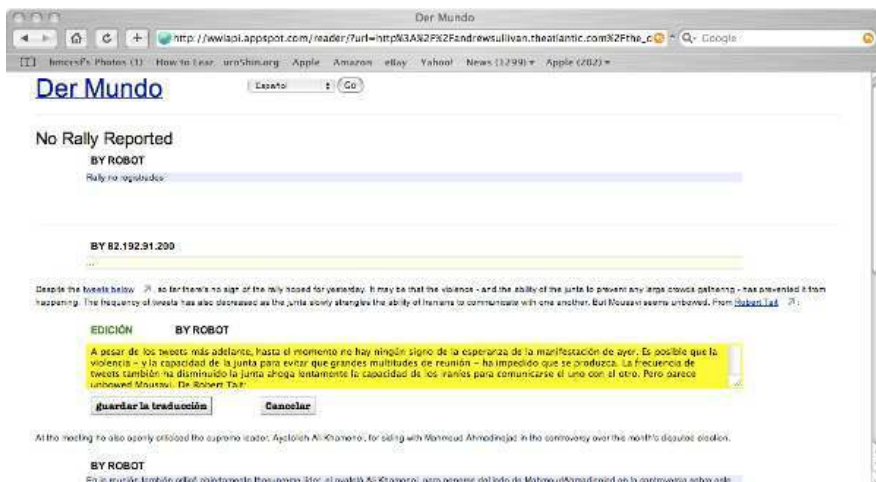
There is a collaborative translation memory that can be embedded in almost any website. It is a hybrid machine/human translation system that combines machine translations from multiple sources with human translations from volunteers or professional translators. It is an adaptive, self-learning system that collects feedback from users to learn which translators submit good or bad work.

The translation memory communicates with the websites using it via a simple web API (application programming interface), and is typically invisible to users who simply visit websites that use WWL as they normally would, and see translations without leaving those websites.

What is it Used For?

Worldwide Lexicon is used for translating content online in a combination of machine and human translation. The system is typically implemented by embedding lightweight widgets, usually written in Javascript, within another website or webpage. The widget loads itself, and then calls WWL to request translations for the texts within the page, and then displays the translations either adjacent to or as a replacement for the original, untranslated texts.

The user will typically see the translations in a dual language view, with a paragraph of original text, followed by a translation, and so on. The user can edit or score a translation simply by clicking on it, editing the text within a text editor form, and then saving it to the translation memory. All of this can be done without leaving the original webpage.



Screenshot : Editing a translation using the embedded javascript translation viewer/editor

Worldwide Lexicon gives you a lot of control over how to display translations, and what parts of a page can be translated. You can translate the entire body of a document in a single block, or you can break it down into smaller units, such as paragraphs, which are each translated independently. You can also allow users to edit translations for some parts of a page, but not others, and you can mix several source languages on the same page.

Resources:

- Worldwide Lexicon Home Page: <http://www.worldwidlexicon.org>
- WWL Documentation: http://www.worldwidlexicon.org/?page_id=9
- WWL Frequently Asked Questions: http://www.worldwidlexicon.org/?page_id=7 sp;

Worldwide Lexicon is free software licensed under BSD.

ADVANCED TECH ISSUES

- 61. Transliteration
- 62. Machine translation
- 63. Translation Memory
- 64. Fonts and Encodings
- 65. Web Fonts
- 66. User Generated Scores

61. TRANSLITERATION

Transliteration is the process of converting text from one script to another, for example from the Latin alphabet to Cyrillic. There are several types of transliteration, some which are based on simple letter substitution, and others that are based on phonetic transliteration, where the goal is to preserve the sound of a word in another language.

The written text in one script does not necessarily have sufficient detail to transliterate into another script. In most cases it is not possible to losslessly transliterate from the Arabic script into the Cyrillic script, as the Arabic script does not usually write short vowels, where the Cyrillic one does.

62. MACHINE TRANSLATION

The field of machine translation (MT) is many decades old, and in fact dates back to the early days of the computer industry. In this chapter, an overview is provided of how machine translation systems function, the use of machine translation and its limitations.

The basic concept of machine translation is to automatically translate a source text in one language into another language. As there are hundreds of widely spoken languages, the number of potential language pairs is astronomical. For this reason, machine translation systems tend to focus on a relatively small number of "high traffic" language pairs where there is demand for translation, such as English to French, French to German, etc.

The demand for translation, not surprisingly, roughly mirrors major world trade routes. The first lesson is that machine translation is only an option for relatively few languages, mostly major European and Asian languages. Open-source machine translation projects, Moses and Apertium, will change this, but building machine translation systems takes work, so this will not happen overnight. This is an important point to be stated up front, that machine translation is not a magic technology. It takes a great deal of effort, either in the form of writing linguistic data or collecting training data to build these systems.

Fortunately, machine translation solves a big problem for governments, world trade, etc. and as such has received a considerable amount of funding over the past several decades. Systems currently in use today, such as Systran, trace their development back for many decades. In the United States, the military and intelligence agencies fund machine translation research, as a way to multiplying their ability to see and understand what is happening overseas. In Europe, the creation of the EU has resulted that governments and private companies have made their laws and trading practices more homogeneous. In general, better communication equals better trade, so in Europe translation efforts have been focused in this area. The majority of translation engines in existence today trace their origins back to one of these two sources.

MACHINE TRANSLATION TECHNOLOGY

In all but the most basic cases, translation is not a simple algebraic process where sentences can be translated according to predefined rules. It demands that the translator, human or machine, understand what is being said, which is often not said, in order to produce a high quality translation. It uses either predefined rules or statistical patterns derived from training data to calculate how a source text may appear in another language. However, it does not understand what it is translating any more than a calculator understands the meaning of pi. Human language is a way of describing a situation or environment, one that includes the full range of human senses, emotion, etc. This explains the limitations of the technology, which will be covered in more detail later in this chapter.

Machine translation today falls into one of three broad categories: rule-based translation (sometimes referred to as transfer-based machine translation), example-based machine translation and statistical machine translation. Rule-based translation is based on the idea that languages have a fairly small set of basic rules, such as how to conjugate verbs, a larger set of exceptions to the basic rules, and a vocabulary or dictionary. Statistical machine translation is based on a different approach. It compares directly translated, or aligned, texts to detect statistical patterns. Phrases that usually translate a certain way show up in many different texts, so the system learns that "white house" usually translates to "casa blanca" in Spanish.

In academia, a large body of text is called a *corpora*. In machine translation, a database of common translations (e.g. "white house" to "casa blanca") is called a *Translation Memory*.

Rule-based translation

Rule-based machine translation is based on the idea that to make a translation it is necessary to have an intermediate representation that captures the *meaning* of the original sentence in order to generate the correct translation in the target language. In interlingua-based MT this intermediate representation must be independent of the languages in question, whereas in transfer-based MT, it has some dependence on the language pair involved.

The way in which transfer-based machine translation systems work varies substantially, but in general they follow the same pattern: they apply sets of linguistic rules which are defined as correspondences between the structure of the source language and that of the target language. The first stage involves analysing the input text for morphology and syntax (and sometimes semantics) to create an internal representation. The translation is generated from this representation using both bilingual dictionaries and grammatical rules.

It is possible with this translation strategy to obtain fairly high quality translations, with accuracy in the region of 90% (although this is highly dependent on the language pair in question — for example the distance between the two).

In a rule-based machine translation system the original text is first analysed morphologically and syntactically in order to obtain a syntactic representation. This representation can then be refined to a more abstract level putting emphasis on the parts relevant for translation and ignoring other types of information. The transfer process then converts this final representation (still in the original language) to a representation of the same level of abstraction in the target language. These two representations are referred to as "intermediate" representations. From the target language representation, the stages are then applied in reverse.

This all sounds straightforward, but it is a lot harder than it looks. This is because computers are very inflexible, so when they encounter something that is outside their programmed rules, they don't know what to do. Examples of the types of things that can confuse a computer are:

- Words slightly out of order, e.g. "I think probably that this might be a good idea". A computer will be totally confused about what's what.
- Words that have many meanings depending on context, e.g. "The boy was looking for a pen, he found the pen, it was in the pen." Humans can usually easily understand the context of a word or phrase, a computer on the other hand does not have the encyclopaedic knowledge available to humans.
- Colloquial expressions, shorthand, and other common types of speech
- Typographic errors.

So now, in addition to writing software to capture every rule in a language, you also have to write software to deal with exceptions to these rules, which can include anything from conjugating irregular verbs, to dealing with common typographic errors. This requires a lot of effort, so not surprisingly good rule-based translation systems exist for a relatively small number of language pairs because of the up front time and money cost of creating all of this information, it can take up to six months to create a basic rule-based MT system from scratch.

Statistical machine translation

Statistical machine translation, developed more recently as computers and storage became cheap, is based on the idea that if you feed a very large number of directly translated texts (known as aligned texts), a computer can detect statistical patterns for how phrases are translated. The advantage of this approach is that the training process is automatic. If you have a large enough training corpus (typically at least 30 million words of aligned sentences), the system can train itself. The important difference with statistical translation is that it is a "blank slate". The computer has no pre-programmed rules or assumptions about how a particular language is constructed. This is both a strength and weakness. It is a strength because languages are filled with exceptions and unusual phrases. Once a statistical system learns how a phrase translates to another language, it does not need to know why. This is a weakness because it has no internal knowledge of the rules of a language, rules which can guide it when it can only translate part of a text. The most popular form of statistical machine translation currently used is phrase-based statistical machine translation, and that is what is described here.

The simplest way to explain how a statistical translation engine trains itself is as follows. Let's imagine that we have a corpus of a million sentences that have been translated from English to Spanish. The translation engine will crawl through the entire set of sentences, breaking each down into smaller blocks of words called n-grams. The sentence "Hello my name is John" could be divided into 4, 2 word n-grams (hello my, my name, name is, is John). This step, breaking texts down into smaller units, is the kind of repetitive task that computers do very well at. Typically these systems will break sentences down into blocks of several different word lengths.

This is where we get into the statistics of statistical machine translation. What the system wants to find are n-grams that are strongly associated with n-grams in another language. Let's say that of the one million training sentences, 1,000 of them contain some version of "Hello, my name is John". Given enough data, the system will learn that 'my name is' is strongly correlated with 'nombre es' in the Spanish translations. The system does not need to understand these words in any way, it is just counting the number of times each n-gram appears in aligned texts, and based on that calculating how strong the correlation is.

So let's imagine that we've trained the system with 1,000 variations of "Hello my name is ____". Then we give it the expression "Hello, my name is Dolores". It has not been trained to translate that sentence yet, but it has seen many other examples. It will break this sentence down into n-grams, and for each one, it will find the best matching translation. It will be highly confident that hello = hola, and my name is = mi nombre es, so it will probably translate this as "Hola, mi nombre es Dolores". You can see the logic of this approach. It works well when you have a large enough training corpus, with some important limits. Statistical translation, like rules based translation, has its weak points, which include:

- Requires a large set of aligned texts, usually millions of sentences to produce adequate results, these training corpora usually require a lot of work to prepare, and are not available for the overwhelming majority of the world's languages
- Very sensitive to text mis-alignment, where source texts are slightly out of sync with translations (a single error in a line break, for example, can ruin the alignment for every sentence pair that follows)
- Potential for information loss when translating between languages with different rules for things such as formality or cases.

The basic concept is that, given a large enough training corpus, you can stitch together composite translations by treating sentences as collections of smaller units. This is a simplified explanation, but it should get the idea across.

STATISTICAL VERSUS RULE-BASED TRANSLATION

Each system has distinct advantages and disadvantages depending on the languages used, application, etc. The short answer, in comparing the two, is that statistical translation can produce very good results, when it is trained with a large enough training corpus. For languages where that is the case, English to Spanish for example, there is a large amount of training data to use. The problem is that for other language pairs, it is very difficult to find enough data, at least in machine readable form, with which to train the translation engine.

Rules based translation systems require a lot of curating, to program the engine with a language's core rules, most important verbs, and vocabulary, but they can also be developed procedurally. Rules based engines are also a good choice for closely related languages where the basic form and vocabulary is not radically different. Apertium, an open source rules based engine, has done a lot of work building translation engines for related languages including Spanish, Catalan, Basque and Galician.

If you have a sufficiently large translation corpus, statistical translation is compelling, but for most language pairs, the data simply does not exist. In that case, the only option is to build a rule-based system.

LIMITS OF MACHINE TRANSLATION

The limit of machine translation is simply that computers are calculators. They do not understand language any more than a calculator understands the meaning of pi. Human language is a form of shorthand for our environment, physical and emotional states. With a single sentence, you can paint a picture of what is happening and how someone responds to it. As you comprehend the sentence, you picture those events in your mind. When a computer translates a sentence, it is simply processing strings of numbers.

The strength computers have is that they have an essentially infinite memory with perfect recall. This enables them to do brute force calculations on millions of sentences, to query a database with billions of records in a fraction of a second, all things that no human can do. Intelligence, however, is much more than memory.

Machine translation will continue to improve, especially for languages that are heavily translated by people, because these human translations can be continually fed back into translation engines to increase the likelihood that when they encounter an unusual text, someone somewhere has translated it. That, in turn, is the fundamental limit for a computer. A person can build a visual picture of what is described in a scene, and if that description is incomplete, he can make a decent guess about the missing part. *A computer doesn't have that capability.*

With this in mind, machine translation is a great tool for reading texts that would otherwise not be accessible, but it is not a replacement for human translators, just as you would never expect a computer to write a novel for you.

MACHINE-ASSISTED TRANSLATION SYSTEMS

Machine assisted translation systems solve this problem by using machine translation where it is strongest, for quickly obtaining rough draft translations that in turn, can be edited or replaced by people. The workflow in this type of system is usually something like the following:

System obtains and displays a machine translation when no human translation is available

- Users can edit or replace the machine translations, either via an open wiki-like system or via a closed process (trusted translators only)
- The system displays human translations in place of machine translations
- Users may be able to score, block or highlight translations based on their quality, which enables the system to make additional decisions about which translations to display

There are several ways of implementing systems like this, depending on the type of application, languages needed, budget, etc. Among the options are:

- Crowd-sourced or wiki translation systems, where anyone can edit and score translations
- Managed translation systems, where only approved translators and editors can edit translations
- Mixed systems where submissions may be open to the public, but are screened by other users or editors prior to publication

In many cases, such as with Google Translator Toolkit, the translations that are given back to the machine are stored in a Translation Memory (TM), so that if the same phrase is translated again the machine translation algorithm already has a reference of how a human did that particular translation. The fact that Google does not share this aggregate data back with the public is bad news for humanity.

FREE AND OPEN-SOURCE TRANSLATION ENGINES

Until recently, all of the machine translation systems in use were closed, proprietary systems (e.g. Systran, Language Weaver). Users were dependent on vendors to support and maintain their platforms, and could not make their own modifications. Fortunately, the open source model is spreading to machine translation. These systems can be modified as needed to accommodate new languages, and will enable small, independent teams to create a wide variety of translation engines for different languages and purposes.

Apertium (<http://www.apertium.org/>) is an open source, rule-based translation engine. It is being used to build translators for closely related languages (e.g. Spanish and Catalan), and also to build translators for less common language pairs, where the data required to do statistical translation simply does not exist (for example Breton to French, or Basque to Spanish). The software opens the door for groups of amateur and professional linguists and programmers to build translation engines for almost any language. This will make rule-based translation available to both smaller commercial opportunities (less widely spoken languages), as well as scholars and hobbyists (e.g. translators for Latin).

Moses (<http://www.statmt.org/moses>) is an open source statistical translation engine that, like Apertium, enables people to build their own translation platforms. The system is aimed mostly at research into statistical machine translation. The engine can be used to make a machine translation system for any pair of languages where there exists a large enough parallel corpus.

The Worldwide Lexicon Project (<http://www.worldwidelexicon.org>) an open source translation memory, has also developed a multi-vendor translation proxy server that enables users to query many different machine translation engines (both open source and proprietary systems) via a simple web API. Most of the Worldwide Lexicon code is released under a BSD-style (permissive) license. For more information on Licensing, see the section under Intellectual Property.

FURTHER READING

- Doug Arnold (2003) "[Why Translation is Difficult for Computers](#)" in Harold Somers (ed.) *Computers and Translation: A translation guide* (Benjamins Translation Library, 35)
- Erik Ketzan (2007) "[Rebuilding Babel: Copyright and the Future of Machine Translation Online](#)" *Tulane Journal of Technology & Intellectual Property*, Spring 2007. Available at SSRN: <http://ssrn.com/abstract=940041>

63. TRANSLATION MEMORY

A **translation memory**, or **TM**, is a database that stores paired segments of source and human translated target segments. These segments can be blocks, paragraphs, sentences, or phrases. Translation memory does not apply to word level pairs, which are covered by translation glossaries. A translation-memory system stores the words, phrases and paragraphs that have already been translated and aid human translators. The translation memory stores the source text and its corresponding translation in language pairs called “translation units”.

Translation memories are typically used in conjunction with a dedicated computer assisted translation (CAT) tool, word processing program, terminology management systems, multilingual dictionary, or even raw machine translation output.

Research indicates that many companies producing multilingual documentation are using translation memory systems. In a survey of language professionals in 2006, 82.5 % out of 874 replies confirmed the use of a TM. Usage of TM correlated with text type characterised by technical terms and simple sentence structure (technical, to a lesser degree marketing and financial), computing skills, and repetitiveness of content^[1]
[http://en.wikipedia.org/wiki/Translation_memory]

Translation memory management involves looking into the TM files for translation that are identical or similar to the ones that we wish to translate. Translations found in the TM database can be identical to the sentence that want to translate, in which case we consider that we have a 100% match for our string, and that the translation should probably be good enough for the string that we want to translate.

EXACT MATCHING VERSUS FUZZY MATCHING

We can also find in our TM translations in which the source string is similar to the source string that we wish to translate, but not identical. In this case we qualify how different the TM string is in terms of percentage. We might find translations that are 95%, or 70% similar to the string that we want to translate, in which case we say that we have a 95% or a 70% match. If we use such strings, they will always have to be considered as fuzzy, and they will have to be reviewed by the translators.

LOCAL VERSUS NETWORK TRANSLATION MEMORIES

Translation memories started out as locally hosted and managed programs (they have been in use since the early days of the personal computer era, before the notion of Internet services existed). As the Internet has developed and broadband connectivity has become ubiquitous, these services are migrating to network based SaaS (software as a service) offerings. Each approach has its down advantages.

Locally hosted translation memories are not dependent on an external network server and can scan the contents of its corpora, on local disk drive or system memory, and can do so very quickly. The burst capacity of a typical desktop computer is quite substantial, so these programs can implement highly compute intensive fuzzy matching algorithms that are difficult to implement on network based systems, and that do not scale as easily in an offsite environment (because of latency from the network connection, disk or data store access, and other performance issues).

Network based translation memories have another advantage because they can be accessed via a web browser or simple client application (e.g. a Java app), and make it easy for many people to work via a single shared translation memory. This approach makes more sense when there is a larger, distributed translation team, such as in a translation service bureau. The main weakness these applications have is that it can be difficult to configure the servers so they are accessible outside an organization's firewall. More and more vendors are offering pure web based translation memories, where all of the data is stored and indexed in a publicly accessible server or server farm.

CORPORA

In linguistics, the term *corpus* (plural *corpora*) is used to refer to a large and structured set of texts, generally kept in a database for research purposes. A corpus is a set of linguistic data. It is important to see that TMs can make useful linguistic corpora, and that massive corpora in the form of TMs are most useful for statistical machine translation (SMT). In both academia and the commercial translation industry, there are many TMs and corpora that are kept private. These are closed corpora. Closed corpora are counter-productive to the goals and ideals of Open Translation.

VENDORS AND TOOLS

Babylon (\$)

[Babylon](#) is a commercial software suite that includes a combination of dictionaries, translation memories and machine translation services. It is not free or open source, but it is inexpensive, well designed and provides very broad support for multilingual dictionaries, domain specific glossaries, and other tools that professional translators need.

Google Translator Toolkit

Part of [Google Translate](#), this is a web hosted translation editor with an integrated translation memory that can be used as a stand-alone translation editor, and also to enable teams of translators to work collectively on a document and to share their translation memory locally or with the web at large. The tool was launched in June 2009 to good reviews. The consensus view among translation experts is that its primary purpose is to backfeed professional quality translations to train their statistical machine translation engine. GTK is not an open source tool per se, as it is a closed system run by Google, but it is free and easily accessible via the web. Google does reserve the right to charge for the service at a future date.

SDL / Trados (\$)

Trados is a commercial software and the most widely known and used TM suite of tools in the professional translation world. This is utilised both by the Translation Agencies and the translators as well. It is a desktop based tool but can be adapted to large CMS as well. SDL is one of the largest translation/localisation companies in the world.

Omega-T

[Omega-T](#) is a free, Java based translation memory that runs as a local, desktop application. It is similar to translation memory tools like Trados.

Worldwide Lexicon

The Worldwide Lexicon is a free, open source suite of translation memory and translation management tools that can be used to create a wide array of multilingual web applications and publications. One of its core features is an open translation memory that combines human and machine translations, and is accessible via a simple web services API. WWL is open source (BSD style license), written primarily in Python and runs on Google's App Engine grid computing platform.

THE TOWER OF BABEL

The rest of the world insisted on character sets more appropriate to their needs, with appropriate punctuation, currency symbols, accented letters, or even a completely different alphabet or other writing system. Thus, the computer industry entered a Tower of Babel period in which user communities and companies developed their own character sets in profusion. See http://en.wikipedia.org/wiki/Character_set#Popular_character_encodings for a partial list.

European languages required additional characters beyond the basic ASCII character set, to display "français" in its proper form "français" for example. Asian languages have much larger character sets to represent thousands of different logographic characters. Their countries created a similar profusion of Double-Byte character sets, such as Big-5 (Taiwan), GB2312 (PRC), HKSCS (Hong Kong) Shift-JIS (Japan), EUC-KR (South Korea).

The variety and disorganization of character set definitions has created numerous problems for software developers and users. The problem began with the common assumption that a program would only be used with one character set. This might work for an accounting program that only complies with the laws of one country, but it has become increasingly untenable in word processing, databases, and so on.

The first reaction was to localize, that is, to create different versions of software using different character sets, with different languages in the User Interface. This is too much work both for developers and for those managing the variety of products that results. It also fails to satisfy the requirements of multinational organizations where individuals may need to work in two, three, or four languages, and the organization as a whole in dozens.

The problem with this is that it greatly complicates things for software developers because you typically needed to install special fonts to display these characters. It was also very difficult to display symbols from different character sets in the same display, which made it difficult or impossible to build bilingual and multilingual interfaces to display many source and translated texts in the same user interface or window.

Encodings and Transformation Formats

A Character Encoding is a mapping between numeric code points and the letters and symbols of a given Character Set. Before Unicode, an encoding was also a mapping to a binary representation of the numeric code point, but in Unicode a code point is a number independent of any representation, and the various possible representations are called Transformation Formats.

We can understand this more easily in human terms. The number ten can be written as the English word "ten", or in various other forms, such as the familiar '10'. It is also '1010' in binary, '012' in octal, '0A' in hexadecimal, "十" in Chinese, "dix" in French, and so on. But it is always the same number. If we number the Latin alphabet A=1, B=2, and so on, are we assigning numbers to the letters, or are we assigning a written form to them? When we start to do ciphers using arithmetic on the letter values, it is clear that we mean numbers. Adding 1 to a letter value, with Z wrapping around to A, is a familiar example. It turns 'HAL' into 'IBM'.

It is important to understand that in Unicode, a code point is a number, not the digital representation of a number. A set of rules for representing code point numbers in binary defines a Transformation Format. For most character sets, where there is only one Transformation Format, it is easy to confuse the two ideas. In Unicode there is one encoding, but many Transformation Formats. The Chinese GB18030 standard is a different encoding of the Unicode character repertoire with the same set of Transformation Formats.

Non-Unicode encodings are classified, misleadingly, as Single Byte, Double Byte, or Multibyte. All 7-bit and 8-bit character sets have Single Byte encodings. Character sets that switch between 16-bit representations of Chinese characters and 8-bit representations of alphabets are said to have Double Byte encodings. Character sets that define a fixed-length representation for characters of two bytes or more are said to use Multibyte Encoding. The variable length UTF-8 form for Unicode fits into none of these three categories, and is described in Unicode terminology as a Unicode Transformation Format, not an encoding at all.

UNICODE

Unicode contains a superset of existing character set repertoires, with each character at a separate code point. It was developed to create one giant character set that could represent every conceivable combination of alphabets, syllabaries, logographic symbols, and special characters.

A single byte character set can represent, at most, 256 unique symbols. Two bytes can represent up to 65536 characters. Three bytes or more can represent many millions of characters. The largest estimates of characters that might get into Unicode are less than 250,000.

The architecture of Unicode was originally set to provide a 32-bit code space, enough for several billion characters. It has since been redefined to consist of 17 pages of 65,536 code points each, for a total of 1,114,112 code points, at least four times larger than necessary.

Bytes / Word	Address Space	Example Systems	
1	256 symbols	ASCII, ISO Latin codes	
2	65,536 symbols	Shift-JIS	
3	16,777,216 symbols		
4	4,294,967,296 symbols	Unicode UTF-32	

Unicode code points are simply numbers. We can represent numbers in a computer in many different ways for different purposes. It turns out that there are several formats that various organizations prefer for one reason or another. The following table illustrates the major formats, using the character U+1D11E MUSICAL SYMBOL G CLEF as an example.

Name	Size, bytes	Endianness	Example
UCS-4/UTF-32	4	LE, BE	0x1E D1 01 00, 0x00 01 D1 1E
UTF-16	2 or 4 (with surrogate pairs)	LE, BE	0xD8 34 DD 1E, 0x1E DD 34 D8
UTF-8	variable	None	0xF0 9D 84 9E

CHARACTER ENCODING (WHY UTF-8 IS YOUR NEW BEST FRIEND)

Unless you have a *very* good reason to use another encoding, your website, application, etc should use UTF-8 as its default encoding. UTF-8 is a variable length encoding system that works in concert with Unicode, and represents each symbol with a word ranging from 1 to 4 bytes in length, with the upper bit in each byte serving as a control register. This encoding method enables an application to encode standard western character sets very efficiently, and to step up to 2, 3 and 4 byte word lengths only when extended characters, pictographic symbols etc are required. This encoding standard also makes it possible to combine strings from many different alphabets and symbol sets in a single string, something that was difficult or impossible with older character sets.

Backward Compatible

UTF-8 is backwards compatible with ASCII characters, and the basic Latin character set. It was designed so that the first 128 code points map directly to the legacy ASCII character set. The system was designed so that code points would map to existing ISO and CJK (Chinese/Japanese/Korean) character sets, which makes porting applications to use Unicode/UTF-8 fairly easy (and trivial for ASCII/Latin applications).

Future Proof

UTF-8, in combination with Unicode, works with a much larger symbol space, up to 4 bytes per word, with up to 28 useful bits per word, for a total symbol space of 268,435,456 unique symbols, far more than are used in every alphabet and written language system in the world. In fact, the system can represent not only conventional writing, but also mathematical and musical symbol systems, and can be extended to represent new symbol sets as the need arises.

Issues For Software Developers

Beware - most programming languages have a significant ASCII "legacy" and will revert back to ASCII encoding, sometimes for no apparent reason. Python, the primary language used in Google's grid computing system, is a good example of this. Python is an excellent language. It is easy to read, but also a powerful object oriented language. It's one major weakness is that it has an annoying habit of defaulting back to ASCII, and doing so inconsistently. When this happens, and you do something like try to concatenate an ASCII string with a UTF-8 string, it will sometimes generate an error. The documentation on how to deal with this is rather poor, so you can spend a lot of time trying to troubleshoot something that should be handled for you automatically.

This issue exists in some form in many programming languages that have Western origins, so you should spend some time understanding how your preferred programming language deals with character encoding and be meticulous about using UTF-8 for string operations, as a default encoding, etc. One notable exception to this rule is Ruby, which having been conceived in Japan, had to deal with multilingual characters from early on. Most major programming languages bolted Unicode support in later in their development.

UNICODE COMPATIBLE FONTS

Many commonly used fonts are NOT Unicode compliant. When designing software and web services, you should take care to use Unicode compatible fonts wherever possible. The standard fonts (Arial, Courier, Times Roman, etc) are usually available in a Unicode-ready form. Fancier fonts, on the other hand, may be limited to Western scripts, and should not be used for multilingual applications. While users can sometimes override the font selection, and sometimes their system will do this for them, it can cause a very confusing situation where the application works normally, but displays garbage characters or empty boxes in place of the current symbols. As of this writing, the following fonts are known to work correctly with Unicode (although they may not support all alphabets and symbol sets, especially for less common languages like Khmer).

Windows

- Arial (included with Windows)
- Arial Unicode MS (included with MS Office)
- Lucida Sans Unicode
- Microsoft Sans Serif
- Tahoma
- Times New Roman

Mac OS X

- Lucida Grande

NOTE: Mac OS X in general has been extensively localized and translated to dozens of languages and geographical regions worldwide. Unicode compatibility on Mac applications and web browsers has not been a widely reported problem, although certain alphabets and symbol sets are often omitted from some fonts.

Other

You can find a more detailed list of free and open source fonts at en.wikipedia.org/wiki/Unicode_typefaces

What Do I Do If The Standard Fonts Don't Work For My Language?

You can try the list of Unicode type faces (via Wikipedia below) as a starting point. Another good strategy is to search on the following term:

```
"unicode font [name of your language or symbol set] [your operating system]"  
for example...  
"unicode font kanji android"  
"unicode typeface kanji android"
```

Even if you speak an obscure language, there is a good chance that a graphic designer has created a font for your language. Some of these fonts are commercially licensed, but many are available as freeware or as open source utilities.

MORE INFORMATION

ASCII (<http://en.wikipedia.org/wiki/ASCII>) : American Standard Code for Information Exchange

CJK (<http://en.wikipedia.org/wiki/CJK>) : Chinese/Japanese/Korean Character Sets

ISO-8859-1 (http://en.wikipedia.org/wiki/ISO_8859-1) : Extended Latin Character Set)

Unicode (<http://en.wikipedia.org/wiki/Unicode>) : Wikipedia

Unicode Compatible Typefaces (http://en.wikipedia.org/wiki/Unicode_typefaces) : Wikipedia

UTF-8 Tutorial (<http://en.wikipedia.org/wiki/UTF-8>) : Wikipedia

65. WEB FONTS

If you are translating text that will be displayed on a web page, and your language uses something other than the latin-1 character set (Roman letters plus a few accented ones), you are probably already thinking about how to display the text so that your readers will see it. You can specify a font-family for use, via CSS markup, but the version of the font your user has may not have support for your language's character set. Worse still, the user may not have the font you specify, and it's going to be a total crap-shoot whether their fallback font has your character set or whether the user will see question marks or little boxes instead.

You can of course provide instructions to the user on how to download a font that will show your language correctly, and many web sites do this. But as of Firefox 3.1, you can also use CSS3 tags to provide a direct link to the font for download via the browser. Only fonts that are actually displayed on the page will be downloaded, so you needn't figure out which fonts are actually used and encode only those in CSS tags.

First, you define the font family by means of the `font-face` tag:

```
@font-face {
  font-family: DejaVu Sans Web;
  src: url(DejaVuSans.ttf) format("truetype");
}
```

The font family name should be a name that is not the name of a font that is widely installed, if you want the user to be forced to download your version of the font. If the user has a local font with the same name, the user's copy will be used for display instead.

Now you can reference this font family anywhere that you normally would in your CSS markup. For example,

```
body {
  font-family: DejaVu Sans Web, Lucida Grande, sans-serif;
}
```

Don't forget to put fallback font faces into your markup; older browsers won't support this tag and they ought to see something specific instead of whatever the user default may happen to be. This ups the odds that the user gets readable text. Note that fonts can be somewhat hard on the dialup connection, so you may want to be economical in your use of multiple typefaces.

Opentype fonts can be included in a similar fashion:

```
@font-face {
  font-family: STIX General Bold Web;
  src: url(STIXGeneralBol.otf) format("onetype");
}
```

One complication is that not all current browsers support all types of fonts, and older browsers have varying to no degree of support for the `font-family` tag. A detailed discussion of this issue is outside of the scope of this book; please see [where?] for more information.

You may want the browser to download the font only in the case that the user does not already have a local copy. (This presumes that the local copy has support for your desired character set.) In this case you will want to add a "local" stanza to the `src` description, like this:

```
@font-face {
  font-family: DejaVu Sans Web;
  src: local("DejaVuSans"),
       url(DejaVuSans.ttf) format("truetype");
}
```

More examples of use of the `font-face` tag are available on the Mozilla blog: <http://hacks.mozilla.org/category/font-face/>.

66. USER GENERATED SCORES

The simplest system is a user-mediated system where users vote on documents, translations and other users. The idea is to give users a means of measuring interest or quality. There are several ways to prompt users and count votes, among them:

- Simple binary voting systems (up/down, plus/minus, good/bad)
- Subjective quality scale (1 to 5 stars, drop down menus)
- Implicit voting systems (number of times viewed, average time spent viewing a page)
- Surveys

Binary Voting Systems

The simplest form of a user based voting system is the binary system, where users are asked to answer a simple question giving two options, such as "Is this a good translation [yes/no]". This can be presented to the user as an icon based interface (up arrow, down arrow), or as a question. The choice of the interface will depend on the visual design of your system. While an icon based interface may seem like a better choice, it can be ambiguous about what a yes vote means, whereas a well phrased question will make it clear that you are asking the user specifically about the quality of a translation.

The data logged in this type of voting system will consist of the following fields, all of which are easy to store in a typical SQL or key/value data store:

- date/time stamp
- unique ID, hash key or serial ID of the translation edit being scored (this ID should be unique for every edit or instance of a translation for a document, so scores are linked to a specific edit, and can be tracked back to the translator)
- voter IP address
- voter location (city, country, lat/long coordinated), derived from IP address via geolocation service such as MaxMind
- voter username (if logged into a registration system)
- vote (+1 or -1)

From this raw data, you can easily compute summary scores (e.g. an average of all scores, standard deviation to measure variability, etc). The summary statistics can be generated in a batch process based on a schedule, on demand, etc.

Subjective Quality Scale

A subjective voting system asks users to score texts on a continuum, ranging from poor (0 points) to excellent (5 points). If users are properly trained, this allows you to collect an extra dimension of data, specifically about the skill level of each user. The binary system is designed primarily to identify very good or very bad or malicious contributors, and does not distinguish between a translator who is proficient in a language but obviously not a native speaker. With a scaled based scoring system, you can associate meaning with a score, such as:

- 0-1 points : very poor quality, malicious edits, spam, should be discarded
- 1-2 points : poor quality, comparable to bad machine translation, should be discarded
- 2-3 points : fair quality, comparable to a decent machine translation, keep it if there is no better alternative
- 3-4 points : good quality, better than machine translation, proficient human translator but not native speaker
- 4-5 points : excellent quality, college level writing, native speaker skill level

The user interface for a system like this is pretty straightforward, as you can prompt users to submit scores via 0 to 5 star buttons or a similar method. Netflix is a good example for this type of voting systems. The main issue one has to be aware of in context with this system is that users can easily become confused and think that you are asking them to vote on how interesting the source material is or other things, which are not directly related to translation quality. Therefore, it is important to embed prompts in the interface so that it becomes very clear to users that this tool is used to assess translation quality rather than how interesting the source document or overall topic is. The data logged by this system is essentially the same except that the voting field is a numeric (integer or float) value instead of a boolean yes/no vote.

Implicit Voting and Scoring

There are also a number of ways you can assess quality and level of interest in translations via implicit web analytics methods. Among the techniques that can be used are:

- Tracking page views for translations across different versions (good translations are more likely to be viewed and shared than others)
- Tracking the time spent viewing a page by reloading an 1 pixel image periodically within a page.

This tracking technique will not give you an absolute or direct way of measuring quality, but you can usually derive from user behavior how good or interesting a text is compared to other pages or URLs in your system. This is related to the technique of using the Internet and web analytics to track links into your system, page view statistics, etc.

The number of page views is not an especially useful way of assessing translation quality because page traffic can be driven by many factors not related to translation quality or even the quality of the source material. Time spent viewing a page is, on the other hand, is a good signal of document quality, although you can't really make out whether users are staying on a page because the underlying subject is interesting, or because the translation is good. It is probably a combination of both. On the other hand, if most users abandon the page immediately, this is a strong signal of a quality problem that may be more strongly associated with translation quality.

You should not use this data to directly set or modify scores for translators, but you can use this data to identify problem areas of your site. If you see a topic or cluster of pages that have very low time on site statistics, this is a sign of one of the following conditions that editors should examine more closely:

- The page is malformed or otherwise unusable or unreadable
- The underlying source document is not interesting
- The translation is very poor, a malicious edit or spam

Think of this as a general purpose quality assurance tool for your website, translation portal, etc rather than as a direct measure of translation quality.

Surveys

Online surveys are a useful tool to ask specific questions to your visitors, but rather about your website as a whole than about an individual person or translation. Examples of questions you might ask people include:

- On average, how would you rate the quality of volunteer translations on this site (1-5 stars)?
- On average, how would you rate the quality of the source pages or articles on this site
- How interesting are they?
- What would you like to see in a more detailed way? (free form question, category multiple choice, etc).
- What type of material are you most interested in translating? (free form question, category multiple choice, etc).

Again, this is not something you can use to directly evaluate individual translations or rate translators, but it is a good way to read the mood of your audience, and get a more detailed information about the type of translations or content they are interested in reading or contributing.

Framing Questions and User Interface Design Issues

Sometimes it does not turn out clearly whether they are voting on how interesting the original article is, or whether they are voting on the translation itself. No matter how clear the symbol is, you need specific explanations that frame the question in each language you plan to quiz users in. Tooltips are a good way to provide additional prompts on top of a visual interface (e.g. up/down arrow icons). The tooltip should clearly explain what you are asking the user to vote on, for example:

"Please rate this translation, NOT the article or content. Rate the translation from 0 (very poor / spam) to 5 (excellent / native speaker)."

The important thing is to clearly state what the user is being asked to vote on, and if you are using a subjective quality scale, be sure to give an unambiguous definition of what different scores mean (e.g. 5 means excellent / native speaker, while 4 means very good / proficient / not quite native quality). As various users will perceive each threshold differently, they should be properly briefed to allow you to collect useful statistics.

For a binary voting system, the question should be framed in a different way, and the votes should be linked to a particular system action. For example, an UP vote might also add that translator to your list of favorite translators, so you will see more of this specific translator's work in the future. On the other hand, one or several DOWN votes may add a translator to your blocked list, so that translations of this translator will be blocked from your view. This is especially useful as a way of detecting and filtering spam submissions.

Voting Privileges and Weighting of Votes

You may decide to combine a user generated scoring system with an expert driven scoring system. In this type of hybrid system, you collect the scores of two different user populations: ordinary or anonymous users, and known trusted users or editors. You can then evaluate the scores submitted by each type of user, using a formula such as:

$$\text{composite score} = (\text{editor score} \times 0.70) + (\text{user score} \times 0.30)$$

This type of weighted score thus treats expert scores as more significant than user scores. You can adjust the weighting ratio according to your own preferences.

Assigning Credit (and Penalty) In Collaborative Translations

As a general rule, you should collect votes on the same unit of texts as translators work on. If your system is paragraph based, scores should be collected for paragraphs, not the document as a whole, because many translators may have contributed to it. It is also important to design a good revision tracking system so that the system obtains a memory of every edit and action within a translation, and not just of the most recent changes.

It is impossible to accurately evaluate units of work, because a translator who only changes a few words, may have contributed significantly to the quality of the whole work, while another translator may rewrite a large block of text, but not significantly change the end result. The best thing to do is to assign a score to whoever touched the item being scored most recently. If you collect enough votes, the distribution between major and minor edits will lead to an average value.

Analyzing Statistics

Calculating raw scores, averages and standard statistical measures (e.g. standard deviation) are easy to do on demand from your raw score logs. If you have a large and dynamic user community, you will probably want to perform additional statistical analysis, both to drive automated rules engines (e.g. which submissions to allow or block), and to learn more about your user community. Examples of the types of statistics you may want to generate are:

- Regression plot of translator scores over the time / age of account, this will tell you if a translator or a translator population tends towards better or worse quality seen over a longer period of time (e.g. translators are learning and improving with growing practice)
- Geographical distribution using geolocation data associated with user IP address
- Temporal distribution : when and how quickly are translations streaming in when an article is published, what is the typical time frame distribution, is there a relationship between source article age and translation volume or quantity?
- Detecting unusual geographical (location), network (IP address / block) or temporal voting patterns, which are a signal for fraudulent or robotic voting activity.

The outputs taken from these processes can be human readable reports and graphs for editors and managers, as well as machine readable reports that can be fed back into rules and workflow management issues (for example, to automatically compute the optimum cutoff threshold of quality scores for allowing or rejecting translations in a peer reviewed system).

Filtering Submission Based on Quality Scoring

Once you have obtained raw and derived statistics from sufficient votes, you may want to feed the data back into your translation management system, either by using summary information to design system rules, or by automatically generating rules based on the statistics you generate. Examples of the types of rules you can generate in this way can be:

- If a translator receives $> x$ DOWN / NO votes from independent users or IP addresses, BLOCK this translator from display to ALL visitors
- If a translator receives $> x$ UP / YES votes from independent users or IP addresses, highlight them as a featured translator and display their submissions first
- If a translator has received fewer than x YES or NO votes, or has a standard deviation $> y$, treat them as unknown and require peer review for their submissions first
- If the system monitors a burst of votes from a certain location, IP address range or within an unusually short time period, IGNORE all votes submitted during this time period or from that source (assume these votes are suspect and possibly robotic).

There is no limit to the variety of rules you can generate from the statistics you collect. In general, for smaller translation communities you won't have a large database of raw votes, so you'll probably want to use simpler scoring criteria and rules, while for a large translation community you will be able to collect a lot of data and learn a great deal about your user community.

Incentives, Penalties and Community Dynamics

Add some text on how incentives promote good work, while negative social pressure can deter incompetent or malicious individuals from returning. Also discuss balancing positive peer pressure with negative pressure, depending on the mood you want to set for translators.

Summary

User mediated scoring systems, when properly designed, are simple to implement and can collect a large amount of raw data that can be further analyzed to develop a picture of who your users are, what they are interested in, and who is doing good or bad work. This also increases participation in the system because individual users can see how their voting activity influences what they see, while translators have a reward incentive to do good work, as this will increase their score and reputation within the system. In all volunteer systems, reputation itself is a form of currency because translators will often use their profiles and by-lines to promote themselves, the work they do, and their companies.

EXPERT DRIVEN VOTING SYSTEMS

In an expert driven voting system, editors or other privileged users are able to vote on translator submissions via a separate administrative interface. This can be done in place of or in addition to user related scoring systems. As your editors and trusted users have a wider knowledge of your system, standards and practices, you can ask them using a different set of scoring criteria. Examples of the types of questions experts can vote on might be:

- Assign quality scores for several dimensions of translation quality (e.g. one score for grammar, one for syntax, another score for general style and a different one for quality of writing).
- Allow editors to accept or reject a large number of submissions in bulk via an administrator's web interface for bulk actions.
- Experts can be trained to use consistent criteria for assigning different quality scores. (Users can do this, too, but may lack the training and experience to do so in a consistent quality).

The data you collect from experts can be used independently or as a composite with user related scores to decide whose translations to accept, who are the best translators, and so forth. You can also generally assume that these users are to be trusted to take decisions without extensive cross-checking (such as a decision to ban a user for submitting spam or inappropriate material).

Framing and Standardizing Questions For Editors

As with ordinary users, it is important to frame questions correctly for your editors. This is rather a matter of training and documentation than it is one of web interface design. You will typically provide a simple set of voting tools that are part of an administrator's interface. This interface will be more utilitarian, so these users can work in bulk, and process large volumes of submissions.

The important thing is to document what the process is for scoring users, what the criteria and thresholds are, and how to perform common administrative tasks. This document should cover topics such as:

- On a 5 point quality scale, which skill levels or quality are required for a given score level (e.g. 4.5 to 5 points means excellent / native speaker, while 4 to 4.5 might mean excellent but not quite native level).
- How do you ban or block a user, IP address or a whole address range?
- How do you perform bulk actions, such as accepting 50 translated texts in a single batch for publication?
- How do you perform common administrative tasks?
- What is the editorial basis for rejecting translations or banning a user?

This is primarily a matter of documentation, as well as providing an online forum where editors and supervisors can meet to discuss items, pending texts or have general discussions related to the system, translations, etc.

HYBRID USER / EXPERT SYSTEMS

A hybrid system combines statistics from ordinary users and editors. There is a variety of ways to do this, including:

- Generating composite, weighted scores that represent both user votes and votes from editors or supervisors.
- Implementing rules where editor decisions may override system defaults (e.g. an editor manually bans or blocks a user or IP address block from the system)

Composite Scores

If both, ordinary users and editors are being asked to submit similar quality scores (e.g. a five point scale), these scores can be combined to generate a weighted score using the formula below:

$$\text{composite_score} = (\text{average_editor_score} \times \text{editor_weight}) + (\text{average_user_score} \times \text{user_weight})$$

where

editor_weight is a weighting factor ranging from 0.00 (0%) to 1.00 (100%)

user_weight is a weighting factor ranging from 0.00 (0%) to 1.00 (100%)

and editor_weight + user_weight = 1.00

This is a simple weighting formula. However, you may want to use different weights depending on how many votes users have submitted, the logic being that a large set of user votes will be statistically normal, and not easily skewed. In this case, you may use a weighting formula that adjust the weights, that is to say, when there are a large number of user votes, the user generated score has a weighting factor closer to 1.00, and when there is only a small number of user votes, the editor's score will have a higher weighting factor. There are countless variations on this theme.

Deciding Which Rules Take Precedence

Another issue to consider in hybrid scoring systems is which generated rules take precedence. To give an example: when should an editor's decision to ban a user take precedence over users' decisions to UP vote or allow the same user. You will probably make the following assumptions in defining these policies:

- Editors are more trustworthy, so his decision to allow or ban/block a user overrides the others
- Large numbers of user votes, if they are not artificially generated, may be more valuable than subjective opinions of individual editors, especially if the goal is to assess quality from the reader's perspective (the editors may be language geeks who downvote a user for minor errors that readers ignore)
- Small numbers of user votes are less statistically significant, and easily skewed, so in situations where an editor score diverges from a user score that was calculated from just a few votes, you may want to assign more weight to the editor's score.

INTERNET BASED SCORING

The Internet itself is a valuable tool for measuring how interested users are in your content and translations, and for detecting major quality or system problems via standard web analytics. This technique is not generally useful for assessing the individual quality of translations, but will tell you things such as:

- Which source articles or translations are most popular in general or within a particular topic, domain, or language
- The "useful life" of a source article and its translations (how quickly does the content age)
- Average time spent looking at the site and at individual pages
- Browser capabilities, language preferences and other visitor information
- How linked is an article, both in terms of inbound links to the source article, as well as third party links to the translated versions

This data, in turn, will tell you what people are most interested in (individual pages, categories, etc), which languages they are reading, translations they prefer or are searching for, where you have major quality problems (very low time spent viewing a page or section), where and how people are visiting your site, and so forth. This will enable you to make editorial decisions about where to direct translators to spend their time, where you focus on marketing your site or service, and so on.

One way for using this technique to detect bad translations is when you are routinely translating source texts to multiple languages. In this case, you may see a significant difference in time spent viewing the various pages, e.g. Japanese users abandon a page within 20 seconds, while French users spend several minutes viewing the French version of the text. This is an indication of a significant difference between two translations which may be due to quality reasons, presence of spam, wrong presentation, etc.

SUMMARY

The mere existence of a voting system is a huge incentive by itself. It enables users and editors to cross check each others, and creates the basis for a rewards based system (or penalty system) that drives the community towards increased participation and a higher quality level of the work. As you can see, there are several techniques that can be used in parallel to assess source text and translation quality, and derived from that, the quality of each user's contributions to the system. In considering these different methods, we'll highlight every method and its relative strengths and weaknesses.

User Mediated Scoring Systems

User mediated systems work best when you have a large user community looking at a steady stream of source texts and the respective translations. It is then relatively easy to prompt readers to score source texts (do they merit translation?) and translations, providing enough votes to develop an accurate picture of what people want, and where the good and weak actors are. The system's main strength is its simplicity, both in terms of what it asks of users, and how the data is stored and analyzed. Its main weakness is that it requires a large number of votes per user system-wide to generate useful reports.

Expert Driven Scoring Systems

Expert driven scoring systems are useful both as a fail-safe (to override bad or damaging contributions or decisions from users), and as a way to compensating for a low volume of user submitted scores (which may often be the case for smaller or less engaged user communities).

Internet / Web Analytics

Internet and web analytics are generally useful for learning who your users are, what they are reading or translating, which languages they read, and where they are spending time. This will help you to understand on which kind of content they are most interested in (and where to direct your translators to spend their time), and where you may have problems within your site (e.g. your Japanese readers are abandoning pages quickly while French users linger around far longer). They generally will not evaluate translation quality, at least not directly, but play a major role in understanding your user community, how people are finding you, and so on.

APPENDICES

67. GLOSSARY

68. License

69. Preparing Content for Translation

67. GLOSSARY

ASCII

ASCII (American Standard Code for Information Interchange) is one of the early character encoding systems for computers. It is a 7 bit, 128 character system that was designed to represent the Latin alphabet, numerals and punctuation. It is not designed to represent characters from other alphabets. This often causes problems because many programming languages were originally developed for ASCII, and only later added support for Unicode and other character sets.

ATOM

ATOM is a content syndication standard, similar to RSS, which allows websites to publish feeds that allow other sites, news readers and web servers to automatically read or import content from each other.

See also RSS.

BRIDGE LANGUAGE

A bridge language is a widely spoken, international language, such as English, French or Spanish, that is used as an intermediate language when translating between two less widely spoken languages. For example, to translate from Romanian to Chinese, one might translate first from Romanian to English, and then English to Chinese because few people speak Romanian and Chinese directly.

See also interlingua.

CHARACTER SET

A character set can be as simple as a table that maps numbers to characters or symbols in an alphabet. ASCII, for example, is an old system that represents the American alphabet (the number 65 in ASCII equals 'a', for example). Unicode, in contrast, can represent a much larger range of symbols, including the large pictographic symbol sets for languages such as Chinese and Japanese.

CHARACTER ENCODING

Character encoding is a representation of the sequence of numeric values for characters in text. For many character set standards, there is only one coding, so it is possible to confuse the two ideas. In Unicode, on the other hand, there is one numeric value for each character, but that value can be represented (encoded) in binary data of different lengths and formats. Unicode has 16-bit, 32-bit, and variable length encodings. The most important is UTF-8, which is to be used for all data transmission, including Web pages, because it is defined as a byte stream with no question of size or byte order. Fixed-length formats also have to specify processor byte order (Big-Endian or Little-Endian).

CMS (CONTENT MANAGEMENT SYSTEM)

A content management system is a piece of software that manages the process of editing and publishing content to a website or blog. A CMS enables editors to supervise the work of writers, manage how articles or posts are displayed, and so on. These systems also make it easier to separate content production (writing) from design related tasks, such as a page layout. [Word Press](#), Movable Type, [Drupal](#) and Joomla are examples of widely used content management systems.

CORPUS

A corpus (plural corpora) is a large and structured collection of texts used for linguistic research. In the context of translation tools, a corpus consist of one or more aligned texts. These corpora typically contain texts that are about a certain domain and consequently can help to find the terminology used in a domain.

COPYLEFT

Copyleft is a use of copyright law to enforce policies that allow people to reprint, share and re-use published content without prior written permission from the author. Copyleft licences require that derivative works use the same licence, so that they are as Free as the original work.

COPYRIGHT

Copyright is a form of intellectual property law giving the author of a work control over its use, re-use in different media, translation, and distribution.

CREATIVE COMMONS

[Creative Commons](#) is an organization that was founded to promote new types of copyright terms, also known as copyleft. The organization has developed legal templates that define new policies for sharing and distributing online content without prior knowledge or consent from the original producer.

DISAMBIGUATION

Disambiguation is the process of determining or declaring the meaning of a word or phrase that has several different meanings depending on its content. The English word "lie", for example, could mean "to recline" (I need to lie down), or "to tell a falsehood". Machine translation systems often have a very difficult time with this, while it is an easy task for humans, who can usually rely on context to determine which meaning is appropriate.

DISAMBIGUATION MARKUP

Disambiguation markup is a way to embed hints about the meaning of a word or phrase within a text, so that a machine translator or other automated process can understand what the author intended. For example, the expression "<div syn=similar>like</div>" would tell a text processor that the word like is synonymous with similar, information a program could use to avoid misinterpreting like as "to like someone".

ETHNOLOGUE

The principal database and catalogue of human languages, providing linguistic and social data for each language. In particular, Ethnologue lists estimates of the number of speakers of each language in each country and worldwide. It is available in printed form and on the Internet at <http://www.ethnologue.org>. Ethnologue's database includes information on more than 6,900 known languages, and continues to grow.

FLOSS

Free, Libre and Open Source Software. An umbrella term for all forms of software which is liberally [licensed](#) to grant the right of users to study, change, and improve its design through the availability of its [source code](#). FLOSS is an inclusive term generally synonymous with both [free software](#) and [open source software](#) which describe similar development models, but with differing cultures and philosophies.

FUZZY MATCHING

Fuzzy matching is a technique used with translation memories that suggests translations that are not perfect matches for the source text. The translator then has the option to accept the approximate match. Fuzzy matching was meant to speed up translation however there is a greater risk of inaccuracy.

GETTEXT

gettext is a utility, available in several programming languages, for localizing software. It works by replacing texts, or strings, with translations that are stored in a table, usually a file stored on a computer's disk drive. The table contains a list of x=y statements (e.g. "hello world" = "hola mundo").

GNU / GPL

GNU or *GNU's Not Unix*, is a recursive acronym for a set of software projects announced in 1983 by a computer scientist at MIT named Richard Stallman. The GNU project was designed to be a free, massively collaborative software, open source software initiative. In 1985 the Free Software Foundation was founded and took up the GNU project. In 1989 Stallman drafted a legal license for his software and called it the GPL or the GNU Public License. The GPL, a copyleft license, is the most popular license for free software.

INTERLINGUA

An interlingua is a artificial language with extremely regular grammar that is used as an intermediate step when translating from one human language to another. This is an alternative to machine translation systems that translate the original text to an intermediate machine representation such as a parse tree, and then to the target human language.

The artificial language Interlingua is sometimes used as an interlingua in this sense. Several other artificial languages, including Esperanto, Loglan, and Lojban, have been proposed for the same purpose.

LANGUAGE CODE

A language code (see ISO) is a two or three letter code that uniquely identifies a human language. For example, en = English, while es = espanol / Spanish. There are two different code sets in widespread use. ISO 639-1 is a two letter code that represents several hundred languages, most of the widely spoken languages in use today, while ISO 639-2 and ISO 639-3 is a three letter code that represents a much larger set of languages (several thousand languages).

LICENSE / LICENSING

Licensing is the process of adding a legal license to your copyrighted work. This copyrighted work may be either a piece of content that can be translated or a software tool for translation. For more information on licensing, please see the chapter on it under Intellectual Property.

LOCALE / LOCALE CODE

A locale code, which is usually a suffix to a language code, provides additional geographical information. For example, Spanish varies by country, so you would identify Mexican Spanish as es-mx, while

Argentine Spanish would have the code es-ar, where the suffix is the two letter ISO country code.

LOCALIZATION

Localization is the process of translating and culturally adapting the prompts, instructions and user interface for a software application or web service. Most applications have dozens to hundreds of system menus and prompts that need to be translated.

MACHINE TRANSLATION

Machine translation is the computerised process of automatically generating a translation of text from one language to another.

MACHINE TRANSLATION (RULES BASED)

A rules based translation engine tries to analyze a sentence, break it down into its parts of speech, and to interpret and disambiguate vocabulary to transform it into an intermediate, machine readable form. It then re-generates the intermediate form into the target language.

MACHINE TRANSLATION (STATISTICAL)

A statistical machine translation system works by sifting through extremely large sets of parallel or aligned texts (sentences that have been directly translated by humans from one language to another). With a sufficiently large training set, or corpora, it learns which phrases are strongly associated with counterparts in the other language. When translating texts, it works by breaking a text down into smaller fragments, called N-grams, and searches for the best statistical match into the target language, and generates a translation by stitching these translated texts together.

MICROFORMAT

A microformat is an open data format standard for exchanging small pieces of information.

OPEN CONTENT

Open Content, a neologism coined by analogy with "Open Source", describes any kind of creative work, or content, published under a licence that explicitly allows copying and modifying of its information by anyone, not exclusively by a closed organization, firm or individual. The largest Open Content project is Wikipedia.

OPEN DATA FORMAT INITIATIVE

Initiative aiming to convince software companies to release *data format* documentation and to pass laws that governments can only store user *in an open format*.

OPEN SOURCE SOFTWARE / LICENSING

To make software Open Source means to put it under a licence requiring that the human-readable source code be available freely on demand, with further rights to modify the program and redistribute the results. Source code under these licences is usually made available for download without restriction on the Internet.

Open Source software was originally defined as a derivative of the Debian Free Software guidelines, when Bruce Perens removed references to Debian from the definition. The current version of the definition is at <http://www.opensource.org/docs/definition.php>

Open Source software is very similar to Free Software, but not at all like Freeware, which is provided at no cost, but without source code. Most Open Source software licences qualify as Free Software licences in the judgment of the Free Software Foundation. The term FLOSS is used to include both: Free (as in Libre) and Open Source Software.

OPEN STANDARDS

An open standard is one created in a publicly accessible, peer reviewed, consensus-based process. Such standards should not depend on Intellectual Property unless it is suitably licensed to all users of the standard without fee and without application. Furthermore, open standards that define algorithmic processes should come with a GPLed or other Open Source reference implementation.

OPTICAL CHARACTER RECOGNITION (OCR)

OCR is the conversion of images to text data, using various methods of shape recognition. The OCR software must recognize layout in addition to character glyphs, in order to represent word and paragraph spacing correctly in the resulting text, and if possible, columns and table layouts. Trainable OCR software can recognize text in a wide variety of fonts, and in some cases multiple writing systems. OCR for Chinese characters and for Arabic presents special problems, which have been to a considerable extent solved.

PEER REVIEW

The process of reviewing a document by independent, possibly anonymous reviewers for quality defined by an appropriate professional standard and the requirements of a particular publication. Standards differ widely in different disciplines.

PO FILE

PO files (extension .po), are text files in a specified format, containing source and translated strings used by the gettext() localization system. Typically, you create one PO file for each language or locale that an application has been localized to.

RSS

Really Simple Syndication - a XML standard for syndicating information from a website, commonly frequently updated databases such as news and events websites or blogs.

SEMANTIC NETWORK

A semantic network is a graph representation of words or phrases and their relationships to each other. In a semantic network, a word is linked to other words via paths, with descriptions of how they are linked. It can represent many types of relationships between words, such as: is similar to, is the opposite of, is a member of a set (e.g. "red" belongs to the set "colors").

STANDARD

A standard is defined by an authority or by general consent as a general rule or representation for a given entity.

STANDARDS BODY

A standards body is an organisation tasked with the definition and maintenance of standards, such as the IETF, which governs Internet standards, or the ITU (International Telecommunication Union), which sets standards for telephonic communication systems and networks.

SVG / SCALABLE VECTOR GRAPHICS

SVG is a XML-based open format for resolution-independent vector graphic files, usually with extension .svg. This allows editing, and thus translation, of any `<text>` elements.

TIMEBASE / TIMEBASE CODE

A timebase code is used in video editing and subtitling to indicate where in a video a particular action, caption, etc takes place. The time is typically expressed as an offset from the beginning of the video clip, usually in a hh:mm:ss:ff form, where hh = hours, mm = minutes, ss=seconds and ff=frame number (e.g. 32 seconds, 12 frames into a clip display the caption "Hello World". There are a wide variety of ways this is done, but the basic concept is similar regardless of file format details.

TRANSLATION MEMORY

A translation memory is a database of source texts and their translations to one or more languages, as well as meta data about the translations, such as: who created the translation, subjective quality scores, revision histories, etc. The main characteristic of translation memories is that texts are segmented into translation units (blocks, paragraphs, sentences, or phrases) that are aligned with their corresponding translations. The standard for translation memory exchange between tools and/or translation vendors is [TMX](#), an XML-based format developed by the Localization Industry Standards Association (LISA).

TRANSLITERATION

Transliteration is a systematic conversion of text from one writing system to another. It is not, in general, simple substitution of one letter for another. The purpose of a transliteration may be to represent the exact pronunciation of the original, or not; to indicate word structure and other linguistic attributes, or not; to represent text in a form familiar to the casual user, or not. There are more than 200 transliteration systems for representing Chinese in European alphabets, mostly Latin with some Cyrillic. Of these, only Pinyin is a standard recognized in China.

Changing fonts is not transliteration. There is, however, an unfortunate practice of creating so-called transliteration fonts, which substitute for the glyphs of a writing system glyphs from some other writing system. The practice is unfortunate because it produces bad transliterations even in the best of cases. Should the Korean family name 金 be transliterated Ro, as written, or No, as pronounced? Should the Spanish name Jimenez be transformed to Chimène in French, as happens sometimes to immigrants? It depends.

UNICODE

Unicode is the principal international character set, designed to solve the problem of large numbers of incompatible character sets using the same encoding. Unicode text can contain symbols from many languages, such as Arabic, English, and Japanese, along with Dingbats, math symbols, and so on. While not all languages are covered by Unicode, almost all official national languages are now part of the standard, except for traditional Mongolian script. In addition to encoding characters as numbers independent of any data representation, the Unicode standard defines character properties, Unicode Transformation Formats for representing Unicode text on computers, and algorithms for issues such as sorting (collation), and bidirectional rendering.

UTF-8

UTF-8 is a variable length Unicode Transformation Format that represents text as a stream of bytes. It was designed so that any ASCII text file (7 bits, with the 8th bit set to 0) is also a Unicode text file. This property does not extend to the 8-bit ISO 8859-1 or Windows Code Page 1252 character repertoires. Extended Latin characters require two bytes each, as do several other alphabets. Chinese characters and some other writing systems require three or four bytes per character. UTF-8 is specified as the appropriate form for transmitting Unicode text, regardless of the internal representation used on any particular computer.

WIKI

A user editable website where users are authorized to create pages, and to create and edit content. Wikis range from open systems, where anyone can edit pages, to closed systems with controlled membership and access rights.

WORD / WORD LENGTH

A computer word is a fixed-length sequence of bits, usually the same length as the registers in the processor. Thus 8-bit, 16-bit, and 32-bit words have been common in the history of computing, and other lengths have occasionally been used.

There is an unfortunate tendency to confuse computer word length with a variety of data types, including numbers and characters. This is most often seen in the mistaken notion that a character is a byte. Even during the period when all character set standards specified 7-bit or 8-bit representations, this was incorrect. Any byte could in fact represent dozens of characters, depending on its interpretation according to a particular character set definition. The idea became more wrong in the case of double-byte character sets for Chinese, Japanese, and Korean, where most characters had 16-bit representations. It is completely untenable in Unicode, where characters can be represented using 16-bit elements (including Surrogate pairs), 32-bit elements, or variable-length byte sequences, as in UTF-8.

XLIFF

[XLIFF](#) (XML Localization Interchange Format) is a standard format for storing localization data. It is widely used by translation memories and translation management tools as an interchange format.

XML

eXtensible markup language is a system for expressing structured data within a text or html document. XML is similar in structure to HTML, and can be used as an interchange format for exchanging complex data structures between different computers. It is often described as a machine readable counterpart to HTML, which is designed to be read by humans. RSS, ATOM, SVG, and XLIFF are all XML based formats.

68. LICENSE

All chapters copyright of the authors (see below). Unless otherwise stated all chapters in this manual licensed with **GNU General Public License version 2**

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

AUTHORS

AboutTheAuthors

© Allen Gunn 2009

ABOUT THIS MANUAL

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Allen Gunn 2009

Andrew Nicholson 2009

Matt Garcia 2009

William Abernathy 2009

Acknowledgements

© Allen Gunn 2009

SCORING

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Eva-Maria Leitner 2009

ANAPHRASEUS

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Gerard Meijssen 2009

Matt Garcia 2009

Silvia Florez 2009

APERTIUM

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Matt Garcia 2009

TECHNICAL CONCEPTS

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009
Ariel Glen 2009
Gerard Meijssen 2009
Sabine Emmy Eller 2009
Thom Hastings 2009

BhopalSurvivors

© Dharmesh Shah 2009
Modifications:
adam hyde 2009

CMS

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Brian McConnell 2009
David Sasaki 2009

CharacterEncoding

© adam hyde 2009

CloudComputing

© adam hyde 2009
Modifications:
Brian McConnell 2009
Thom Hastings 2009

COMMUNITY MANAGEMENT

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Sabine Emmy Eller 2009

OPEN TRANSLATION

© adam hyde 2009
Modifications:
Allen Gunn 2009
Ar Ah 2009
Brian McConnell 2009
michelle yan 2010
Sabine Emmy Eller 2009

MACHINE TRANSLATION AND COPYRIGHT

© Ed Bice 2009
Modifications:
Thom Hastings 2009

Corpora

© Gerard Meijssen 2009
Modifications:
adam hyde 2009
Wynand Winterbach 2009

Costing

© Dwayne Bailey 2009
Modifications:
adam hyde 2009

CREATING

© adam hyde 2009

Modifications:
Andrew Nicholson 2009

CREDITS

© adam hyde 2006, 2007, 2009
Modifications:
Laura Welcher 2009
Thom Hastings 2009

CURRENT STATE

© Authors Of Open Translation Tools 2009
Modifications:
Allen Gunn 2009
Lachlan Musicman 2009
Sabine Emmy Eller 2009
Wynand Winterbach 2009

CustomSearchEngines

© Dwayne Bailey 2009

DICTIONARIES

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Dwayne Bailey 2009
Eva-Maria Leitner 2009
Gerard Meijssen 2009
Lachlan Musicman 2009

DigitalSupportForLanguages

© Gerard Meijssen 2009

FLOSS MANUALS

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Kim Murray 2009

FINDING

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Andrew Nicholson 2009
David Sasaki 2009
Lachlan Musicman 2009

CHARACTER ENCODING

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Anders Pedersen 2009
Brian McConnell 2009
Edward Cherlin 2009
TWikiGuest 2009

FREE SOFTWARE

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Edward Cherlin 2009

Lena Zuniga 2009

GAUPOL

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Andrew Nicholson 2009

Lachlan Musicman 2009

Matt Garcia 2009

GLOBAL VOICES

© Georgia Popplewell 2009

Modifications:

adam hyde 2009

David Sasaki 2009

Jeremy Clarke 2009

Silvia Florez 2009

GLOSSARY

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Andrew Nicholson 2009

Brian McConnell 2009

Edward Cherlin 2009

Gerard Meijssen 2009

Matt Garcia 2009

Silvia Florez 2009

Thom Hastings 2009

Wynand Winterbach 2009

GLOSSMASTER

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Lena Zuniga 2009

Matt Garcia 2009

History

© adam hyde 2009

FORMATS

© adam hyde 2007, 2008

Modifications:

TWikiGuest 2007

INTRODUCTION

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Gerard Meijssen 2009

Matt Garcia 2009

TOOLS

© adam hyde 2009

Modifications:

Matt Garcia 2009

Internationalisation

© Gerard Meijssen 2009

Modifications:
adam hyde 2009
Dwayne Bailey 2009

INTERPRETING

© adam hyde 2009
Modifications:
Brian McConnell 2009
Dwayne Bailey 2009
Eva-Maria Leitner 2009

INTRODUCTION

© Ethan Zuckerman 2009
Modifications:
A Haris Kartasumitra 2010
adam hyde 2009
adrian hernandez 2009
Allen Gunn 2009
David Sasaki 2009
Ed Bice 2009
Generational Equity 2009
Gerard Meijssen 2009
Laura Welcher 2009
Matt Garcia 2009
William Abernathy 2009
Wynand Winterbach 2009

LICENSING

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Ahrash Bissell 2009
Thom Hastings 2009

Locales

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Brian McConnell 2009
Dwayne Bailey 2009

LOCALISATION

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Dwayne Bailey 2009
Francis Tyers 2009
Gerard Meijssen 2009
Lachlan Musicman 2009
Sabine Cretella 2009
Wynand Winterbach 2009

MACHINE TRANSLATION

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Anders Pedersen 2009
Brian McConnell 2009
Francis Tyers 2009
Matt Garcia 2009

Thom Hastings 2009

MACHINE TRANSLATION

© adam hyde 2009

Modifications:

Dwayne Bailey 2009

Francis Tyers 2009

Thom Hastings 2009

Meedan

© Ed Bice 2009

Modifications:

adam hyde 2009

TWikiGuest 2009

MissingOpenTranslationTools

© Allen Gunn 2009

MobileTranslation

© adam hyde 2009

Modifications:

Anders Pedersen 2009

Brian McConnell 2009

MOSES

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

Matt Garcia 2009

OLPC & SUGAR

© Edward Cherlin 2009

Modifications:

adam hyde 2009

OfflineStrategies

© adam hyde 2009

OKAPI FRAMEWORK

© Yves Savourel 2009

OMEGAT

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Lachlan Musicman 2010

Lena Zuniga 2009

Matt Garcia 2009

OMEGAT+

© Raymond Martin 2010

OnlineStrategies

© adam hyde 2009

WIKIS

© adam hyde 2009

Modifications:

David Sasaki 2009

OpenLinguisticData
© Ed Bice 2009
Modifications:
Thom Hastings 2009

OPENOFFICE
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Matt Garcia 2009

OpenStrategies
© adam hyde 2009

WHAT IS OPEN TRANSLATION?
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Allen Gunn 2009
Brian McConnell 2009
Ed Bice 2009
Edward Cherlin 2009
Matt Garcia 2009
Sabine Emmy Eller 2009
Thom Hastings 2009
William Abernathy 2009
Wynand Winterbach 2009

POFiles
© adam hyde 2009
Modifications:
Brian McConnell 2009

PLAYING
© mick fuzz 2007
Modifications:
adam hyde 2007
Andrew Nicholson 2009
Thomas Middleton 2008

POOTLE
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Matt Garcia 2009
Philip Olson 2009
Silvia Florez 2009

PreparingContent
© Dwayne Bailey 2009

PREPARING CONTENT
© adam hyde 2009
Modifications:
David Sasaki 2009
Gerard Meijssen 2009
Matt Garcia 2009

QUALITY CONTROL
© Authors Of Open Translation Tools 2009

Modifications:
adam hyde 2009
Brian McConnell 2009
Gerard Meijssen 2009

Regional/Cultural Issues
© Allen Gunn 2009

REPUTATION METRICS
© adam hyde 2009
Modifications:
Brian McConnell 2009
Gerard Meijssen 2009

SVG
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Matt Garcia 2009

SCRIPTING SVG
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Matt Garcia 2009

SignLanguages
© Gerard Meijssen 2009

ROLES IN OPEN TRANSLATION
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Allen Gunn 2009
Dwayne Bailey 2009
Sabine Emmy Eller 2009

STANDARDS
© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Dwayne Bailey 2009
Francis Tyers 2009
Gerard Meijssen 2009
Yves Savourel 2009

DISTRIBUTION
© Authors Of Open Translation Tools 2009
Modifications:
Andrew Nicholson 2009

FILE FORMATS
© adam hyde 2009
Modifications:
Andrew Nicholson 2009
Lachlan Musicman 2009

SUBTITLES
© Authors Of Open Translation Tools 2009
Modifications:

adam hyde 2009
Anders Pedersen 2009
Andrew Nicholson 2009
Matt Garcia 2009

Terminology

© Authors Of Open Translation Tools 2009
Modifications:
Brian McConnell 2009
Dwayne Bailey 2009
Gerard Meijssen 2009

TikiWiki

© Olaf-Michael Stefanov 2009
Modifications:
adam hyde 2009
Annabelle Hacking 2010
Matt Garcia 2009

Traduxio

© philippe Lacour 2009

TRANSLATE TOOLKIT

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Lena Zuniga 2009
Matt Garcia 2009

Translatewikinet

© Gerard Meijssen 2009
Modifications:
adam hyde 2009
Christopher Lynch 2009
Matt Garcia 2009

TRANSLATING SVG

© Authors Of Open Translation Tools 2009
Modifications:
adam hyde 2009
Matt Garcia 2009

TranslatingSubtitles

© adam hyde 2009
Modifications:
Andrew Nicholson 2009

TRANSLATION

© Allen Gunn 2009
Modifications:
adam hyde 2009
Dwayne Bailey 2009
Gerard Meijssen 2009
Sabine Cretella 2009

TranslationDictionaries

© adam hyde 2009
Modifications:
Gerard Meijssen 2009

THE TRANSLATION INDUSTRY

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Allen Gunn 2009

Dwayne Bailey 2009

Ed Zad 2009

Gerard Meijssen 2009

Sabine Emmy Eller 2009

TRANSLATION MEMORY

© adam hyde 2009

Modifications:

Brian McConnell 2009

Ed Bice 2009

Ed Zad 2009

Lachlan Musicman 2009

Thom Hastings 2009

TRANSLATION MEMORY

© adam hyde 2009

Modifications:

Brian McConnell 2009

Colin Brace 2010

Dwayne Bailey 2009

Gerard Meijssen 2009

Sabine Emmy Eller 2009

TRANSLATION

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Dwayne Bailey 2009

Eva-Maria Leitner 2009

Sabine Emmy Eller 2009

TRANSLATION TIPS

© adam hyde 2009

Modifications:

Ariel Glenn 2009

Brian McConnell 2009

David Sasaki 2009

Ed Zad 2009

Matt Garcia 2009

TRANSLITERATION

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

Francis Tyers 2009

Gerard Meijssen 2009

VIRTAAL

© Authors Of Open Translation Tools 2009

Modifications:

Dwayne Bailey 2009

Matt Garcia 2009

Silvia Florez 2009

WWL

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

Lachlan Musicman 2009

Matt Garcia 2009

WWLHowTo

© adam hyde 2009

Modifications:

Brian McConnell 2009

Matt Garcia 2009

Thom Hastings 2009

WEB FONTS

© adam hyde 2009

Modifications:

Ariel Glenn 2009

INTRODUCTION

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

Christopher Lynch 2009

David Sasaki 2009

Dwayne Bailey 2009

Lachlan Musicman 2009

Matt Garcia 2009

WEB TRANSLATION SYSTEMS

© adam hyde 2009

Modifications:

Brian McConnell 2009

David Sasaki 2009

Francis Tyers 2009

Gerard Meijssen 2009

Matt Garcia 2009

WhatIsTranslation

© Ariel Glenn 2009

Modifications:

adam hyde 2009

Allen Gunn 2009

Edward Cherlin 2009

WHY TRANSLATE

© Authors Of Open Translation Tools 2006

Modifications:

adam hyde 2006, 2007, 2009

Allen Gunn 2009

Ben Akoh 2009

Ed Bice 2009

Edward Cherlin 2009

Laura Welcher 2009

Matt Garcia 2009

philippe Lacour 2009

Sabine Emmy Eller 2009

William Abernathy 2009

Wynand Winterbach 2009

Wikimedia

© Ariel Glenn 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

WIKIPEDIA

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Andrew Pompa 2009

WORKFLOW

© Authors Of Open Translation Tools 2009

Modifications:

adam hyde 2009

Brian McConnell 2009

Dwayne Bailey 2009

Ed Zad 2009

Sabine Emmy Eller 2009

YEEYAN

© Jiamin Zhao 2009

Modifications:

Carolyn Anhalt 2009



Free manuals for free software

GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

69. PREPARING CONTENT FOR TRANSLATION

An attempt can be made to translate almost anything into another language. But if the source text is difficult to understand and translate then it will have impacts that content producers should be aware of. These include, taking longer to translate, the message being lost in translation or not being translated at all.

The translation industry employs a few strategies to help ensure that content can be well translated. These include:

- Constraining language - by limiting the terminology, complexity and style of technical manuals it is possible to ensure that they remain translatable.
- Pretranslation - in this process an editor, who understands the issues of translation into the target languages, makes changes to the source text to ensure that it is translatable.

COMMON PROBLEMS AND SOLUTIONS

The following are a list of the most common issues and how they might be addressed.

Style

The source content may be in various styles, some of which might not work in the target language. A simple example would be where content is in a very personal style while the target language employs a very impersonal style in this type of content.

The source content needs to be adapted to address the issue or the translation brief should specifically state the change in register is allowed in the target languages. In the long term it might be worth establishing a style guide for the source documents.

Complex Sentence

The creator of the source document might make use of a style that creates sentences with more than one key point. A pretranslation editor would break these into two sentences.

Consistent Use of Terminology and New Terms

It is always good to build a terminology list for the domain, this helps the translators when they are translating. In the same way the source document should consistently use that terminology. A pretranslation editor would adjust the use of terms to align with the terminology list.

Any new terms that are found that need definition and that will need to be developed in the target language are added to the terminology list.

Logical Flow of Arguments

In the heat of a blog post an author might make an argument that is poorly developed, that makes a leap of faith or that needs a minor tweak. A pretranslation editor would help to clarify this logic either by correcting it or adjusting it with the author. This ensures that translators are not faced with the issue of having to build the arguments themselves.

Repetition of Logic

An author may repeat the same idea a number of times using different examples or arguing from different directions to arrive at the same conclusion. A pretranslation editor would either merge these arguments into one, ensure that they are each logical or write something to the translators explaining that there are two points being developed.

Foreign Language in the Source Text

Content creators may include foreign phrases, borrowed words, slang and other words or expressions that the translator may not be familiar with. An English author writing in South Africa might borrow Afrikaans or Xhosa words and expressions. The pretranslation editor might remove these or explain their meaning in a general way so that translators can translate them. The editor could build the explanation into the source text so that it is easily translated and give instructions not to translate the original.

Content creators might want to avoid using terms that might be specific to their locale or to always explain words and phrases that could cause confusion. There is of course a balance in that a personal piece full of colour and expression should not become academic or plain.

Idioms, Examples and Cultural References

Idioms can be some of the hardest things to translate as they have many levels of meaning. A translator would need to understand those meanings to be able to find equivalents in their language. This is one reason why many people insist that translation is towards a translator's primary language as it is only in this language that the translator has full access to equivalents. A pretranslator can explain the idiom to the translator or even highlight the key part of the idiom that is being used in the context.

Examples are the easier of this group to adjust. It's often easy to find examples from the target language's locale. Thus the pretranslation editor can either find general examples or allow translators to adjust the example to their locale as needed.

Cultural references would include quotes, movie dialogue, etc. "Play it again Sam", "Open the podbay door Hal", "Beam me up Mr Spock" are all references to popular culture which may or may not be a part of popular culture in the target language. However, the target language might have a rich parallel popular culture for example science fiction culture in Hungarian is very rich thus offering alternatives. The pretranslation editor will choose their approach based on the target languages, these would include asking for a similar reference, explaining the context of the reference or eliminating the reference.