Dictionary Development Workflow for MT: Design and Management

Mike Dillinger

Logos Corporation
100 Enterprise Drive, Suite 501 Rockaway, NJ 07866
USA
mdillinger@logos-usa.com

Abstract

An important part of the development of any machine translation system is the creation of lexical resources. We describe an analysis of the dictionary development workflow and supporting tools currently in use and under development at Logos. This workflow identifies the component processes of: setting goals, locating and acquiring lexical resources, transforming the resources to a common format, classifying and routing entries for special processing, importing entries, and verifying their adequacy in translation. Our approach has been to emphasize the tools necessary to support increased automation and use of resources available in electronic formats, in the context of a systematic workflow design.

Keywords

dictionary development; lexicon; machine translation; workflow

Market forces are pressuring language engineering in general and machine translation in particular to evolve from cottage industries producing fine, handcrafted exemplars of custom systems to lower-cost, quickly-developed, open and adaptable components. There seems to be a clear consensus that language engineering technology is technically feasible; the problem now is to show that is economically viable. Since human intervention plays such an important role in the economics of developing natural language systems, management of human resources for such labor-intensive products will be an important factor in determining the future of the industry.

A significant part of the development of a commercial machine translation (MT) system for a new language pair or a new domain is the development of the lexical resources that the system will use. However, there seems to be little literature available about efficient procedures or "best practices" for developing these resources. Even at the 1993 AAAI Symposium on "Building Lexicons for Machine Translation", there was more attention to theoretical issues than to establishing effective processes for dictionary development.

Lexical resources are available in electronic format, techniques exist for extracting terms from large corpora, and standards are being developed for the exchange of lexical data, but it is clear that there are no *de facto* standards for format or content, no leading providers of lexical data, and no best practices to follow for someone who has to start building a new dictionary today. On the other hand, apocryphal accounts abound of bleary-eyed, underpaid and overworked interns and relatives "just typing the dictionary in". This is clearly not the method of choice for a company that has to deliver a quality product on time, under budget, meeting the customer's needs for lexical coverage, and then scale up to do the same for several languages and several different domains simultaneously.

Since dictionary development can easily consume 30% or more of the total resources allocated for the development a language pair, in a commercial setting it has to be taken very seriously. A systematically designed and tested workflow for the development of lexical resources can lead to reduced expenses, higher quality products, quicker time to market and more reliable outsourcing of lexical development efforts. The present paper reports on-going work at Logos investigating the design and management of dictionary development workflow in the context of commercial MT and explicitly directed at reaching these goals. The general approach taken is to analyze and decompose the process to be able to introduce pre-defined procedures, piecemeal automation and improved tools wherever possible.

Analysis of our experience developing the dictionary for a new language pair (English-Portuguese) and expanding dictionary coverage to new domains led us to identify the following component processes of dictionary development:

- setting lexical goals,
- locating and acquiring resources,
- transforming the resources to a common format,
- classifying and routing entries according to the kinds of special processing they require,
- importing the entries, and
- verifying their adequacy in translation.

We describe the issues in terms of our own system, although the same issues arise for any such development effort.

Set Lexical Goals

It is surprisingly difficult to set relevant goals for dictionary development, since often little more than intuition can guide us in specifying the lexical requirements for an MT system, unless we have the good fortune to work in a controlled-language scenario with a pre-defined lexicon. Rough estimates of 40,000 to 100,000 entries (needed for a general-purpose MT system) help us judge the parameters of the problem, but

contribute little to determining *which x* thousand words need to be included. Since closed-class words will comprise only about 1,000 of these entries (for most European languages), it is clear that the focus must be on the open-class nouns, verbs, adjectives and adverbs, with special attention to nouns and noun phrases, which can make up half of the lexicon, and often more than that.

The workflow management issue here is that without knowing *which* terms are needed *when*, much time, money and effort is spent on processing terms that are simply not being used once they are in the system, while terms that are needed right away only appear in the dictionary much later. A system for identifying and prioritizing lexical requirements will make a significant contribution to solving this problem.

At present, there seem to be three equally arbitrary approaches to setting lexical goals, all of which are apparently in common use:

The first is the "marketing approach": stipulating a number of entries that is comparable with or greater than the numbers cited by the competition's marketing people. The fact that different systems assess their lexical coverage in very different ways (full-form vs. canonical-form dictionaries, for example) makes these numbers meaningless, of course. Even so, this goal is useful for planning and declaring a dictionary effort to be finished.

A second approach is the "resource approach": given a resource such as an existing language pair, a dictionary, glossary or word list, the goal is to get the set of corresponding entries into the system. This has the considerable advantage of specifying just which entries have to be included, although the basis of choice for many resources is questionable. This approach is what we have often used to develop a new target language for an existing source or to put a given glossary into the database.

A third approach is the "sample approach", which is to get into the dictionary all the entries needed to deal with a specific, usually small, sample of texts. Again, this has the advantage of providing an explicit decision about which terms to include and an explicit goal. Depending on the sample size, however, the usefulness of these terms to the system in general is often dubious. We have used this approach when meeting the needs of a sample translation for a prospective client, for example.

The importance of these approaches is not linguistic or technical, since none of them provides systematic indicators of lexical coverage or dictionary quality. The value of such goals is managerial, an aide to monitoring progress and making it possible to declare a project done.

A more systematic approach to setting lexical goals, the one we are developing at Logos, is to determine domain-specific lexical requirements, in the form of a list of frequent or important source-language terms computed over a representative sample of texts in the domain. Such requirements provide a "to do" list but also provide a more adequate indication of lexical coverage or adequacy for the domain in question, that can be used to compare different systems or different phases of the same system more objectively. Our work toward this goal has taken two complementary approaches.

One approach was to search the Web for monolingual glossaries for the domain and collate them in a single document with a uniform format. Merging several of these glossaries yields a list of source-language terms that workers in the domain considered important enough to include in a representative glossary, along with their definitions, which are used later as an aid to finding and checking transfers. Such glossaries usually have a few hundred well-chosen entries.

The second approach also exploited the Web by automatically compiling a domain-specific corpus and then processing that corpus to extract open-class terms. We built a spider using a simple algorithm to collect web documents to form the corpus. Initial results collecting web pages that were returned by a search engine were disappointing, in general because most pages had very little text and yielded few useful terms. A second version of the spider follows the links provided by a search engine and collects only the documents (.doc files, because they usually have much more text than web pages) that are attached to the relevant sites, saving them locally for further processing. This approach is being refined to double check the contents of the files as well, because we have seen that getting one or more hits for a term on a web page often does not correlate well with the terms within the documents that the page links to. In practice, the accuracy of the topic detection is not crucial, since this approach tends to be over-inclusive and non-domain terms will appear with much lower frequency than the important domain terms.

To set goals for lexical development based on a corpus, we need frequency data that is collapsed over the different inflected forms of a word and that ranges over multiple-word terms. The terms identified will specify what is needed to cover the domain, and the frequency data provide a direct indication of the priority of the terms for inclusion. Most word count functions, however, count only single words and specific character strings. To obtain the necessary data, our system has a facility for identifying the terms in a text that are (and are not) found in the dictionary, called TermSearch. Since it uses the word lookup functions and noun phrase analyzer of the MT system's parser, it is much more intelligent than a simple word count and provides exactly the data needed, with frequency data pooled over different inflectional forms. We also have a facility to merge the results of several such TermSearch reports, which provides a list of unfound terms, ordered by decreasing frequency, as a Lexical Requirements Specification for a given domain.

The advantage of guiding dictionary development with a Lexical Requirements Specification is that a more relevant, explicit goal is used, thus reducing drastically the time spent on processing entries that are not immediately (if ever) useful and guarantees that the terms with the greatest impact on quality of translation will be entered first. The corpus building tool allows us to compile a representative amount of text, in a workable timeframe, to assure a reasonably reliable sample. Of course, a Specification constructed by this method is not totally accurate: common words from outside of the domain will be included and uncommon words from within the domain will be left out. It should, however, be

clear that it provides a much more accurate, prioritized guideline than just a number or a glossary whose entries are chosen subjectively.

Locate and Acquire Entries

There are five principal sources of terms for dictionary development: human specialists, parallel corpora, print dictionaries, machine-readable dictionaries, and web-available dictionaries. We are developing strategies and tools to work with each of them.

Human Specialist Knowledge

Sometimes glossaries are not available for a given domain and recourse has to be made to human domain specialists so that they can enter the necessary terms directly. To do so, they need to use a simple, general tool that requires little or no specific training, yet still provides results that facilitate further processing. For this scenario, we use any spreadsheet program to produce a tab-delimited file that can be imported automatically into the system with little need of further processing.

Parallel Corpora

Parallel corpora are notoriously difficult to mine for new terminology in a reliable way. Moreover, the corpora necessary for specific domains and specific language pairs are often not readily available. As the data and techniques become available, we will go on to build tools for dictionary development from this source as well.

Human-readable Dictionaries

Paper dictionaries and word lists are often very useful sources of dictionary entries, although they are slower, more expensive, and more tedious to process because they have to be processed manually by humans.

To facilitate both maintenance of the MT system's dictionaries and this kind of manual work when it is necessary, we have developed a suite of specialized terminology tools, the Logos Language Development Environment, that is accessed via a browser interface over the Web so that many users can work on the same dictionary database simultaneously from different locations. The dictionary is implemented as a robust and scalable Oracle relational database, and the tool provides a graphic interface for searches of arbitrary complexity, as well as for adding, deleting, copying and modifying entries.

Machine-readable Dictionaries

MRDs can be grouped into three classes: those with implicit structure (e. g., a glossary or paper dictionary in .rtf format), those with explicit but not necessarily relevant (for MT) structure (e.g., a dictionary in SGML or XML), and those with explicit structure tailored to the needs of natural language processing (i.e., exchange formats such as OLIF [www.olif.net] or SALT [http://www.ttt.org/salt/]).

Anyone who has tried it knows that parsing a paper dictionary in txt or rtf format is a headache. Formatting conventions are never adhered to strictly enough and they sometimes change from entry to entry. However, it is clear that a few Perl scripts can whip most of the entries into a usable form with much less effort than typing them in or reformatting them by hand.

Explicit structuring makes the parsing and reformatting steps almost trivial and, in the case of exchange formats, virtually guarantees the presence of the information necessary for most MT systems.

The tool suite for our system includes a facility to import OLIF-formatted dictionaries directly; other structured formats can be imported by recasting them as tab-delimited files.

Web-available Dictionaries

There is an increasingly relevant array of lexical resources available via the Web, both general coverage dictionaries and specialized glossaries (for example, from the International Monetary Fund [http://www.imf.org] or the United States Department of the Treasury [http://www.ots.treas.gov/glossary.html]). These are often more up-to-date and more specialized than the paper dictionaries that are available and so provide information that is not available elsewhere.

We can mine several of these public domain resources with a spider that queries these sites, parses the content returned and provides the relevant information in an appropriate format for import or for use by terminologists. Provided with a list of source terms from our requirements list, the spider can go to any of a number of such sites and retrieve the necessary transfers or definitions when they are available. This cuts down drastically both on browsing time for web-based lexical research and on the time spent for locating entries in paper dictionaries. We have found that simply providing candidate transfers (rather than source terms only) regularly doubles terminologists' productivity, since they can focus on choosing the most appropriate transfer.

Reformat Entries

Whatever the source of the entries to be processed, they have to be parsed and reformatted (to a greater or lesser extent) to become compliant with the format we use for our Import facility.

Exploiting web resources today, however, requires a customized parser for each resource, because there are no standards in use to determine what content to include or in what format to display it. We are already readying our tools for work with the Semantic Web (Berners-Lee, et al, 2001; W3C Semantic Web Activity Statement, 2001; Allen, 2001), which foresees rich metadata labeling of web resources for easy automatic identification and parsing of the information in the resource. We can already generate on the fly the system-specific metadata that our system needs, so we'll be able to browse multiple, distributed lexical resources over the web, as they become available to meet the needs of different customers. As these lexical resources appear (and are automatically identifiable as such), MT providers will be able to cut development costs and increase the diversity and quality of the lexical resources they use.

For the time being, however, the first, and sometimes non-trivial, step is to parse the entries in the

lexical resource file. Both the files and entries vary in format from resource to resource. We have built ad-hoc Perl programs for parsing several different resources and are now exploring the use of a parser generator (Conway's RecDescent module for Perl [http://www.perl.com/CPAN-local/modules/by-authors/Damian_Conway/]) to parameterize this process, making it easier to modify for reuse.

Parsing the entries also serves to identify the information available and differentiate (e.g., for print dictionaries) between dictionary entries and example sentences or phrasal entries, which will be processed differently or discarded. Since glossaries and language engineering systems vary very much in focus and implementation, there are almost always different kinds of information in entries of different lexical resources. Therefore, during the reformatting step, we delete any unneeded information and identify any missing information that is essential for import and try to provide it programmatically (e.g., word class or gender) when it is absent from the to-be-imported entries.

Our Import function had already been streamlined so that it requires very little information from the user or glossary. We have made further progress in this area by enhancing the Import function so that it no longer requires the user to specify gender for words in Portuguese, Spanish or Italian. For these languages, only source term, source gender, source part of speech, target term are needed for each entry. Thus our system is less dependent on the variability among external lexical resources and more easily adapted to whatever conventions are adopted by existing resources or for future resources on the Semantic Web.

Since we often process homogenous resources, source language, target language, subject matter and a glossary id can be supplied programmatically. For other languages, gender has to be provided by hand until we finish the gender logic needed for them as well. We also want to enhance this step with automatic spell checking.

Classify and Route Entries

When the set of entries from a given lexical resource has been acquired and formatted for import, they have to be checked and routed. The entries have to be checked against the existing database to see if they are redundant, before any human verification. If they are not redundant, they may also have to be subjected to different kinds of additional processing before import and vetting.

A recent enhancement to our workflow is the automation of this step. Previously, a lexicographer would have to assess each entry for redundancy, validity, and further processing, supplying any further information needed, at the time of assessment. Because, when they were presented in alphabetical order, each entry had different requirements, little batch processing was possible and the lexicographer's work was not as focussed and efficient as we would like. Moreover, the lexicographer spent time assessing entries that only would be deleted as redundant, or laboriously checking each entry's existence in the database.

Now most of the assessment work is automated. For example, entries with non-canonical forms (plural nouns, inflected verbs, etc.), explanations rather than transfers (for now, we do a simple check for length), multiple transfers, or verbs and forms derived from verbs are all gathered and separated for off-line processing. For example, *-en* adjectives are processed differently in our system depending on whether they are deverbal (ex: hidden) or not (ex: wooden) and require a human decision, so we separate them and automatically generate example sentences such as "John has <hidden|wooden> someone" to facilitate the decision process.

This automation allows us to both identify the entries that require special treatment and to forward the unproblematic ones directly to import. The tool is very fast and produces homogenous files that require a single kind of off-line human processing. This provides the human workers with a more focussed task and relieves them of the work of deciding what kind of processing will be necessary. Moreover, it identifies very quickly the subset of entries that need no immediate revision, thereby eliminating the need for human assessment of them. The next step planned for this system is to have it route the entries to different specialists by e-mail, according to the type of off-line processing or review that is needed, and to keep a record of what data sets were sent to whom, when they were returned, etc.: an automated lexical workflow system.

Import Entries

In our system, we only import canonical forms, so importing entries entails expanding the source term to create new entries for any derived forms that the database needs (such as *-ing* forms) and then Autocoding each entry to generate automatically all of the grammatical and database attributes that the system needs, e.g. head of noun phrase, word count, inflection pattern, etc. for both the source term and its transfer. Thus, upon import, we can generate a complete range of system-specific metadata for external lexical datasets, whether glossaries from customers or remotely-hosted, self-labeled resources on the Semantic Web.

Currently, the accuracy for these autocoded attributes is very high, with few exceptions that require human post-editing of the suggestions that the system provides. After this step, the new entries are ready for immediate use in translation. Improvements to this process focus on identifying the head of multiple-word terms, automating a more detailed semantic classification of terms, and labeling the terms that may have less reliable attributes.

Verify Entries in Translation

At a couple of steps in the dictionary development process, as well as at this final step of checking how well the transfers provided behave in translation, the workflow requires human intervention to assess and revise the dictionary entries. As the diversity of the languages and subject matter skills increases, so do the geographical demands made on the system. It is becoming increasingly necessary to have terminology teams in different countries and consultants for different areas in different cities, since no one lexicographer is likely to be familiar with the

terms from the wide variety of domains an MT system can be call upon to work in. A single database with local access would make this impracticable. Multiple copies of the same database in different locations leads to significant problems of keeping them all synchronized and the most recent version available to translation.

Our solution was to develop a browser-based interface (the Logos Language Development Environment) with advanced terminology tools and translation facilities that permits multiple users from any location to access, develop and maintain the same database over the Web as well as use it for translation (in development or in production).

Another dimension of this stage of revision is compiling sentence contexts for testing lexical items. The importance of this is difficult to overstate: for one large technical glossary that we prepared, only 6% of the entries were ever used in translating a half-million-word general corpus. Obviously, this general corpus could only be used to assess the presence of any unwanted side effects on matching the main dictionary entries, but definitely provided almost no information about the new technical entries.

To make this work more efficient, we have been experimenting with concordance tools to extract test sentences automatically for given lexical and phrasal entries from a corpus on disk or from the web in general. Although of course such tools will often not identify the correct sense of the word, we find it helpful to have example sentences to work from and to provide contrastive contexts for different word senses.

Conclusion

We have presented here an overview of the issues involved in designing a systematic workflow for the development of lexical resources based on different kinds of source material. We have also described some of the tools we have developed to support this workflow.

To synthesize, the dictionary development workflow we are developing consists of well-defined steps and procedures that focus on:

Setting lexical goals through the development of a domain-specific, corpus-based Lexical Requirements Specification that identifies which entries are needed and how to prioritize them;

Locating and acquiring entries from a diverse array of paper, machine-readable and web-available lexical resources in as automated way as possible;

Parsing and reformatting these entries to extract the information available and supply any missing information that is needed, again by automated means;

Classifying and routing these formatted entries to eliminate redundancies and group the entries for more focussed and efficient human verification, even by geographically diverse teams;

Verifying the behavior of these entries in translation by generating grouped and targeted test sentences besides the standard corpora that we normally use.

We have already benefitted from the adoption of these procedures and with future efforts to integrate existing tools into a unified, web-based workflow environment, we will be able to develop lexical resources more efficiently than ever before. Our customers will also benefit from these enhancements when they contract our services to customize MT to their needs or when these tools are bundled with our MT system to enhance their own, in-house, dictionary development.

Acknowledgements

Thanks go to my co-worker Frank Gontier, who provided invaluable assistance in implementing various parts of the system described here.

References

Allen, J. 2001. Making a semantic web. [www.netcrucible.com/semantic.html].

Berners-Lee, T., Hendler, J., & Lassila, O. 2001. The Semantic Web. Scientific American, 284(5), 34-43. [www.sciam.com/2001/0501issue/0501berners-lee.html].

W3C Semantic Web Activity Statement. 2001. [www.w3.org/2001/sw/Activity].