Otelo and the Domino Translation Object

Alan Barrett
Lotus Development
Ireland

Abstract

Lotus is working with SAP and a number of MT vendors to make usage of MT easier. Most of this work has been done in the framework of the Otelo project. It is also part of Lotus' efforts to make development of multilingual web applications much simpler. Terminology interchange and text interchange formats as well as a Linguistic Services API are discussed. Also covered is the Domino Translation Object which enables use of these technologies on the Domino infrastructure.

1 Introduction

The lack of access to knowledge because it is only available in languages not understood by people who need that knowledge is a huge economic issue. This applies to knowledge on the web, knowledge needed to effectively transact e-business and to knowledge of wider cultural value. This is particularly an issue for people who only understand "minority" languages. While many senior people in organizations may understand a range of "international" languages, people at a lower level in organizations may be excluded from full participation. Existing and potential customers or business partners may also be excluded.

Lotus is working with SAP and a number of MT vendors to make usage of MT easier. Most of this work has been done in the framework of the Otelo project which has been part funded by the European Commission. However, this paper is not a full discussion of the Otelo project.

Four contributions to MT quality and access to MT are discussed in this paper:-

- OLIF (Open Lexicon Interchange Format)
- OText (Open Text interchange format)
- Linguistic Services API
- Domino Translation Object

2 OLIF

In order to make MT more useful to organizations there needs to be a simple way of importing terminology into the proprietary lexicons of MT engines. A corporation may have a large store of terminology available (e.g. from human translation efforts) which, if easily usable, will make MT much more productive. Additionally, if a corporation has control of it's terminology in it's own repository and it can easily be imported by MT engines then the corporation will probably be much more comfortable with MT (i.e. the corporation's terminology will not be locked into a proprietary format). It will also allow the corporation to use the best MT engine (along with the MT engines own supplied terminology) for any particular job. Open access to terminology preserves an organization's expensive investments in terminology development.

OLIF is a standard XML interchange format open to all. An entry in OLIF consists of:

- Central entry defined on a concept basis (single & multiword units): defining features are canonical form, syntactic category, domain classification/reading & language
- Linguistic description (for each stem variant and also for the entry as a whole) based on what MT systems really use
- Links to other entries such as synonyms and transfers to other languages (including equivalence type and additional tests & actions)
- Terminological description & maintenance information

OLIF has been designed with additional uses in mind such as Cross Language Information Retrieval. It is intended to simplify exchange and sharing of terminology between a variety of technologies.

OLIF is supported by a number of MT vendors including Lernout & Hauspie and Logos. A consortium was founded in early '99 to promote the usage and further development of OLIF and related technologies.

3 Otext

Otext defines the minimum markup set necessary to support exchange of text between different systems. It follows a shadow file approach in that, when a rich text file is converted to OText, the OText file only contains text relevant to linguistic processing. Other source file contents such as graphics are retained in a shadow file. Sufficient information is kept in the OText file so that the source file can be reconstituted from the OText file and the shadow file.

OText is mainly intended as a standard format for MT processing. Most MT systems understand HTML and plain text. However they no not understand, and should not need to understand, a wide range of text formats (e.g. Notes rich text). It is intended that filters will take a wide range of file types and convert them to OText to make the job of MT engines much simpler.

OText is a proper subset of OpenTag. At its simplest it contains text paragraphs with references into the shadow file created by the filter. These paragraphs can contain the following markup:

- Format information (which may also reference more detailed formatting information in the shadow file) and may be duplicated or dropped by the MT system
- Reference place holders (into the shadow file) that maintain their position with respect to the surrounding text and may be moved with that text, but not dropped, by the MT system
- Instructions to protect certain text from MT processing (e.g. text such as a filename may be part of the text flow but should not be translated)

OText is intended to make it easy for filters external to an MT system to pass just the necessary information to an MT system.

Otext is supported by a number of MT vendors including IBM, Lernout & Hauspie and Logos.

4 Linguistic Services API

The Linguistic Services API is a Java RMI interface which is intended as an open way of accessing linguistic services (over a network or locally) from an application.

The primary interface defined by the API is a "gateway". A gateway typically provides access to many different services, although it could simply be a single service such as an MT engine for a specific language pair.

The gateway interface has been designed so that gateway implementors may use an interface exactly the same as the gateway interface as the interface to access their own services. This allows simplicity of implementation of services (because they look just like gateways) and also would allow services (and gateways) from other vendors to plug easily into the higher level gateway.

Of course a gateway implementor may hide all the details of their service implementation behind the gateway if they wish to.

A gateway which provides compute intensive services such as MT will typically run on a dedicated machine. It is possible for gateways to be nested so that functionality may be distributed across a network. Load balancing and scalability can be provided by having multiple instances of the same service on different machines.

A typical application would query a gateway to see which services are available (e.g. for MT which language pairs are supported). It would then configure options (such as which terminology sets to use) and send text for processing (e.g. translation).

Examples of services, other than translation, rnight be language identification, summarization or terminology processing.

Although the linguistic services API is defined in Java it could be used to access Corba objects using IIOP. DCOM might also be supported.

Lotus is working with software companies, gateway vendors and MT vendors to enhance, publish and standardise the Linguistic Services API. This standardization effort is intended to overcome the current problems in the MT market due to incompatibility of solutions from different vendors. Companies working with Lotus on this include Alis Technologies, Inc., Lernout & Hauspie and Logos.

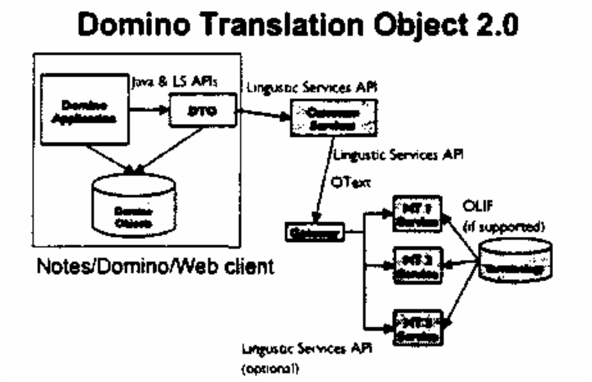
No one MT engine vendor will be able to provide all the language pairs required by customers worldwide. Also MT engines can vary significantly in quality from language pair to language pair. The Linguistic Services API architecture is designed to relieve customers from having to write different application support for each language pair and to allow developers to seamlessly integrate Machine Translation functionalities into their applications.

5 Domino Translation Object

Lotus Domino, among other things, is the premier environment for developing and deploying interactive web applications for Intranets and the Internet. The Domino Translation Object (DTO) allows Lotus' business partners and customers to easily MT enable their Domino applications. Applications using the DTO can be written in Java or LotusScript (Lotus' object oriented VBA compatible scripting language). They can run on a browser (Java only), on the Notes Client or on the Domino Server.

DTO 1.01 has shipped and is supported by gateways from Alis Technologies, Inc. and Lernout & Hauspie. DTO 2.0 is due at the end of the year and will access gateways using the Linguistic Services API.

DTO 2.0 uses the Linguistic Services API to access a common services gateway which provides services such as text filters (to OText). This common services gateway passes on requests for other services (such as MT) to the appropriate gateway.



As well as basic text translation the DTO provides the following functionality:-

- Configuring options such as which terminology sets to use.
- Simple noun terminology support and DNTs
- Requesting translation of more complex data such as some or ail of the fields in a document, or all the documents in a view.
- Call back events when a portion of work done (otherwise methods return when fully complete).
- Error handing.
- Aborting a requested action.
- Providing statistics such as words translated.
- Access to other linguistic services available on any gateway.

Some usages of the DTO might be as follows:-

- Allowing users to casually translate documents. In some cases they then might want to route the documents for human translation.
- Developing a sophisticated translation workflow including revision marking of updated documents, translation memory, terminology development, MT and professional human translation.

Any Domino developer should be able to MT enable their application or write new applications around MT. However, in addition to the business application and supporting technology, for a complete industrial strength solution, our business partners and customers will need access to MT integrators that can customize MT based solutions. In particular ensuring that the appropriate MT terminology sets are extended to cover the application domain. We are working to extend the list of business partners who have specialist skills in these areas.

5 Summary

Lotus has a rich environment for developing industrial strength interactive multilingual web applications. Access to linguistic services is a key part of this.

The MT market can be expanded if there are easy ways for developers to access a wide range of language pairs from different vendors. There is a huge demand for machine translation, if it can only be made easy to use. It is hoped that the initiatives discussed in this paper will be a major step towards building up the machine translation industry and market.

Further Information

http://www.lotus.com/international

http://www.otelo.lu

Lotus, Lotus Notes, Lotus Domino and LotusScript are registered trademarks and Domino and Notes are trademarks of Lotus Development Corporation.

Java is a registered trademark of Sun Microsystems, Inc. All other company names and products are trademarks or registered trademarks of their respective companies.