

Machine Translation as a Global Enterprise Service at Ford

Craig Plesco

Ford Motor Company
Dearborn, MI 48121
USA

cplesco@ford.com

Nestor Rychtycky

Ford Motor Company
Dearborn, MI 48121
USA

nrychtyc@ford.com

Abstract

Ford Motor Company is at the forefront of the global economy and with this comes the need for communicating with regional manufacturing staff and plant employees in their own languages. Asian employees, in particular, do not necessarily learn English as a second language as is often the case in European countries, so manufacturing systems are now mandated to support local languages. This support is required for plant floor system applications where static data (labels, menus, and messages) as well as dynamic data (user entered controlled and free text) is required to be translated from/to English and the local languages. This facilitates commonization of business methods where best practices can be shared globally between plant and staff members.

In this paper and presentation, we will describe our experiences in bringing Machine Translation technology to a large multinational corporation such as Ford and discuss the lessons we learned as well as both the successes and failures we have experienced.

1 Introduction

Ford Motor Company is a global corporation with facilities and people in all developed regions of the

globe and a subsequent need to share and communicate with staff and plant employees in many languages. Efficient best practices can be facilitated in part by learning the successes and challenges of each region and then exploring and sharing effective adjustments to global methods.

Machine Translation (MT) provides the basis for our efforts in these areas, and we have had varying levels of success with supplier and open-source solutions. We are currently supporting 18 languages with over 90,000 translation requests per day via our web interface and 14,000 daily via background batch processing. Most recently we have also deployed an XML over HTTP based web service which acts as an application programming interface (API) to multiple systems (for batch and real-time translations), providing a front-end to two MT supplier engines (Systran and SAIC), wrapped in our pre and post processing routines. The batch and web service modules include many linguistic processes that we have developed internally to aid in pre-translation formalization, expansion of acronyms, and part-of-speech based processing (to help differentiate short phrase semantics), as well as post-translation memory caching and abbreviating (to shorten translations to pre-determined lengths) in various languages.

Online translations available to the general public give the impression that translations are "plug and play" to many business customers, so the user community expects to use MT without any customization work. Our challenges include driving the formulation of industry and business unit specific translations (phrases and acronyms), including development of translations for

languages which are not readily available off-the-shelf (finding, creating, and using parallel corpus data). Some of the recently introduced countries often do not even have developed terminology for manufacturing processes as the industry is new to them. Engaging local users for glossary development and post-editing of results (including the ability for key users to override translations within one application) are important components of this effort. The loop of translations to user overrides/feedback and back will allow our MT system to become more accurate over time.

Managing customer expectations becomes easier when we are able to support business specific terminology in a secured (internal) environment. Success in manufacturing has allowed us to become engaged in other business units (such as product design, warranty, and quality) and to branch out into translating for analytics purposes.

In this paper and presentation we will describe our experiences in introducing Machine Translation technology to Ford Motor Company and discuss the lessons we learned (both positive and negative) as part of this process. Section 2 examines our initial work in developing a controlled language and implementing supplier solutions to MT, Section 3 looks at our extension of MT to the enterprise, Section 4 explores management of user expectations from MT, and Section 5 explores some of our current work in MT pre-processing and plans for error handling in source text.

2 Initial Work

2.1 Standard Language

At Ford Motor Company we have been working with MT in the manufacturing arena since the late 1990s and have had success in integrating and applying natural language processing (NLP) techniques into our corporate systems that manage the vehicle assembly process for all of our assembly plants throughout the world. Our initial application utilized the development of a controlled language (Huijsen, 1998), known as "Standard Language", which was created for use in writing vehicle assembly process instructions.

The goal of developing Standard Language was to make it flexible enough for the Ford process engineers to describe any manufacturing assembly work that needed to be done and to make the language restricted enough so that it could be parsed unambiguously in order to provide the best translations. We have continuously modified and added on to Standard Language by introducing new manufacturing process verbs, new tooling and other terminology to support the very dynamic automobile manufacturing world.

Standard Language is processed using a parser based on an Augmented Transition Network (ATN) architecture (Charniak et al, 1987) and the terminology is stored in an ontology based on semantic networks and description logics, which includes word/concept relationships incorporating base classes, synonyms and various attributes such as part of speech, type, and physical attributes.

Standard Language contains about 5,000 lexical terms and its success can be verified by the fact that it is used within all of Ford's manufacturing operations around the globe. Our experience with Standard Language gave us valuable insights into the application of NLP to real-world usage. Many of the process engineers that used the system were not altogether thrilled with having to use a controlled language that had to be validated by the AI system before their instructions could be sent to the assembly plants and they were very enthusiastic about pointing out errors in our system. Our team did not initially include any people with linguistic knowledge and we added constructs to Standard Language that were grammatically incorrect but supported the business processes.

This use of "non-standard" language was not an issue until we were required to enhance our system to be used at assembly plants outside of North America, where the workers did not speak English. Since we were using a controlled language with a restricted terminology and grammar, we assumed that a machine translation approach could be easily implemented. However, our use of ungrammatical terminology in Standard Language led to problems with the Machine Translation (MT) system. This required us to develop a front-end preprocessor system that would convert our Standard Language into a more "machine translation friendly" format that could be translated with higher accuracy. We realized that cleaning up the text prior to Machine

Translation was a much more effective way to improve the translation accuracy than by trying to change the MT system (Rychtycky, 2007). A working example of this pre-processing is presented here in section 5 (Current Work).

2.2 Integration of Supplier Engines

Our primary process planning system in manufacturing is called GSPAS (Global Study Process Allocation System), and within it was our first major application of MT. Utilizing Standard Language, the process engineers write instructions for manufacturing of vehicles and component products. Many of these process engineers are central staff writing in English based Standard Language, but the instructions do need to be translated to the local languages so that operators know how to assemble the product they are working on.

We integrated our AI and NLP processing with supplier translation engines into a tightly coupled subsystem tasked with translating these instructions to the various global languages required. The tight coupling consisted of a direct connection to the GSPAS database, where new assembly instruction records were identified nightly and translated and stored into new records for regional operations. An outline of this process is provided in Figure 1.

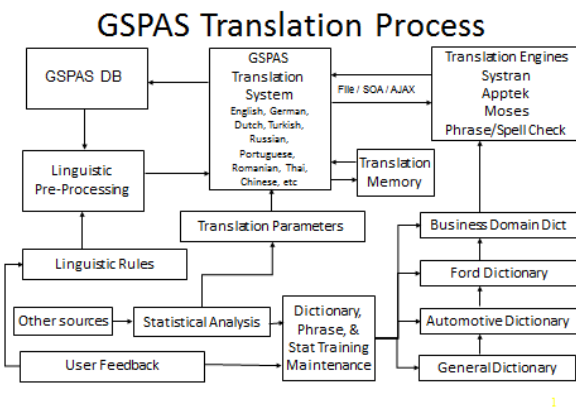


Figure 1: Tightly coupled translation process

The system uses a hierarchy of translation dictionaries starting with a general dictionary from the supplier (Systran, SAIC) and then extended with tools provided by the suppliers to create dictionaries specific to the automotive industry, Ford business, and specific Ford business units.

These dictionaries and related profiles allow us to create translations that are best suited to the terminology and phrases used by different user groups (vehicles, engines, and later extended outside of manufacturing). A translation glossary usually consists of 6000-7000 entries for a language pair.

The GSPAS approach was adequate when translations were focused on this part of the business, but when we began getting inquiries from other application owners and other business units we knew that the approach would need to evolve.

3 Extension to the Enterprise

We realized that extending our translation model to the enterprise would require both an API for other systems to interact with at the system level as well as a manual ad-hoc method for people working with global communications in emails and documents.

3.1 Application centric to service oriented

The first step in extending our model was to change the manner in which GSPAS itself had data translated. We created a web service to encapsulate all of our NLP processing and MT engines, and decoupled the service from the GSPAS database. This created two modules: 1) a module that interrogated the GSPAS database for translation requirements and called the new service for translation, and 2) the new translation service. In figure 2 we show a modification to the GSPAS process so that it becomes a versatile enterprise solution.

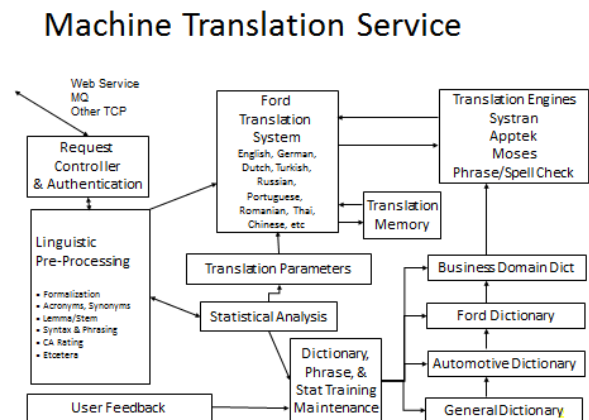


Figure 2: Loosely coupled translation process

The new service incorporates a front-end web service listener that answers on an HTTP/S port and communicates in a pre-determined XML format for requests and replies. The service verifies that the calling system is pre-authorized for translation activities and has a pre-defined profile for translating in the requested language pair. Options for pre-processing include synonym lookup, acronym expansion, abbreviation of terms for translating into a finite space, etc. Communication with the Systran engine is via their own web service facility (SOAP over HTTP), and with SAIC is via their LMTTP interface. A user of our service does not need to worry about communication with the various translation engines as this is handled by our service which acts as a single point of contact for all their translation requirements.

The basic requirements of the service request are the application ID (to authorize along with IP address, and to identify special processing), application keys for each translation (which we hand back with the translation so that they can use an internal and/or meaning pointer to their records and then apply the translation more readily), the text(s) to be translated (up to 100 per XML request), the source language, and the target language (can be overridden at the individual text level). The reply gives back the application keys along with the original source text(s) and the translation text(s). If any errors are encountered (such as language pair invalid or engine services unavailable) these are indicated in a required status ("OK", "ERROR", and message) portion of the reply XML.

3.2 Web Front End

Addressing the ad-hoc translation needs of individuals is handled via a web site that we have developed, based primarily on Systran Enterprise Server (Surcin, Lange, and Senellart, 2007.). The website is presented as a frame within a window where the user can switch between the standard Systran web interface and another page we have developed which allows translations with the SAIC engine (Matusov and Köprü, 2010) via a web form passing data to an intermediate program that handles LMTTP calls.

The Systran web interface allows the end user to create their own sign-in ID where they can generate linguistic option profiles to change their translation results. The site allows for manual entry (or copy/paste entry) of phrases for translation, as well as submission of documents (Word, Excel, Powerpoint, PDF, etcetera) that will be translated by the engine using profiles that we or the end user have created.

4 Managing Expectations

The web savvy user community is aware of external services such as Google Translate that allow them to quickly translate phrases for themselves, however, this can create confusion in expectations.

4.1 Providing and communicating service

Users will compare translations from external websites with ours and comment that sometimes the external site provides "better results." In some cases, this can be true depending upon the context of the text to be translated. If the user translates conversational text in an email using a profile in our system that is meant to address business unit terminology, he can come away disappointed. One of the biggest challenges in providing translations in the enterprise is in managing and directing expectations. Consistent communication about expectations is key to letting people know that context matters, profiles matter, longer phrases can provide better results than shorter, well-structured grammar and syntax matter, and that above all translation services are not a "black box."

As the Ford NLP group, we have defined a process for translation service that manages expectations by stepping through triage of data and requirements and formulation of specific solutions. We first examine sample data with the users and provide feedback on suitability for translation, how we can assist with pre-processing and profile/glossary development, and most importantly the need for human review. Only after this and related steps do we provide access to the web service API for their application. As well, our total process reinforces the notion of human review by requiring maintenance reviews of the

translations with feedback from the end users. This feedback is important so that we can understand and adjust the translation profiles, or even rely upon more exclusively in the case of short phrases (such as 1-3 word menu and label items for an application where context cannot readily be determined in automation).

4.2 User Satisfaction

As an ongoing quality assurance initiative, we conduct user satisfaction surveys on a regular basis. This allows us to pinpoint areas where translation services can be improved, whether it be work on business unit specific profiles and linguistics, or in simply communicating about options and expectations around translations. Figure 3 shows a graph of user satisfaction with translation accuracy.

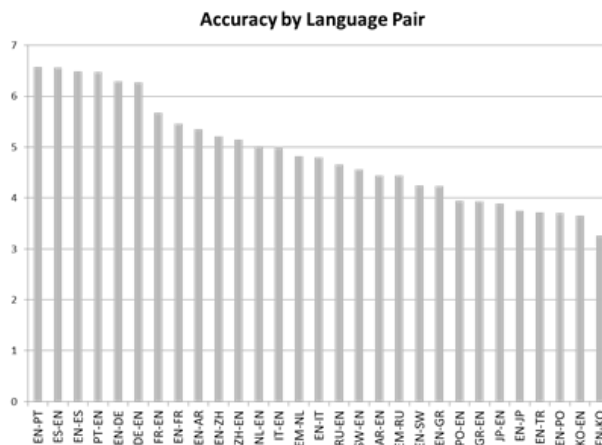


Figure 3: Language translation accuracy

The single most notable result from our surveys thus far is that satisfaction is higher where use and translation feedback (from users to us) are also higher. This underscores the notion that user feedback on translation is critical in deployment of MT in the enterprise if it is to be accepted and embraced by the user community as a tool that they will adopt for business purposes. It is also important to point out that expectations of translations must be reflective of actual business needs. A gist level of understanding a translation can be acceptable in conversational translations, but in other instances such as highly technical documentation the accuracy and quality of the

translation is required to be much higher. Business terminology can be built into the glossaries, but obviously only in tandem with a well engaged user community.

5 Current Work

Controlled languages offer many benefits, as discussed previously, however, they also introduce some challenges. Two tasks in our current work address such challenges: the need to formalize standard language for translation, and analysis of human errors in using something other than their primary natural language.

5.1 Formalization of text for translation

Translated language text can lose or change meaning when the source language text is not specific enough in its meaning or in some cases is too informal. Therefore, we are now addressing this need to formalize Standard Language text to facilitate improved machine language translation by post-processing (after user input has been accepted). This is a good example of the type of pre-processing that we have needed to perform in order to gain quality results from MT.

As one example, here is an in-depth look at our current work related to the insertion of the definite article (THE) into sentences to change noun phrases to the correct determiner phrases, providing a more formal source text to translate. This is a significant challenge because the definite article is normally used as a marker to determine where a noun/determiner phrase is located, but we need to use other means. We assume first that our nouns can use definite articles as the manufacturing process refers to specific and unique objects that have been obtained for a build process.

Our current approach is the addition of a post-processing parse model that traverses in reverse, that is, from the bottom of the syntax tree to the top, and inserts the definite article when reaching an appropriate preposition or verb (assuming an article is not already in place). Support for this approach comes from Heim and Kratzer (1998) where the defined semantic types/functions (where e=entity and t=truth) include:

- Proper names (DP): D_e

- Nouns, Adjectives, Intransitive verbs: $D_{\langle e,t \rangle}$
- Transitive Verbs and many Prepositions: $D_{\langle e, \langle e,t \rangle \rangle}$
- Be/Is and A (identity functions): $D_{\langle \langle e,t \rangle, \langle e,t \rangle \rangle}$
- The (definite article): $D_{\langle \langle e,t \rangle, e \rangle}$
- Quantifiers (including cardinal and "some"): $D_{\langle \langle e,t \rangle, \langle \langle e,t \rangle, t \rangle \rangle}$

As background, determiner phrases are formed by a combination of a determiner (in this case, the definite article) and a noun, giving a semantic type of D_e (individual), as shown in figure 4, where the determiner takes the NP as an argument, which is expressed by the lambda calculus formula $[\lambda f : f \in D_{\langle e,t \rangle} \text{ and there is exactly one } x \in D_e \text{ such that } f(x)=1 \cdot \text{the unique } y \text{ such that } f(y)=1] ([NP])$.

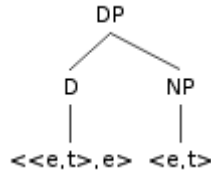


Figure 4: DP semantic formation

Additionally, the individual type D_e (determiner phrase) is normally an argument of only two semantic types: the transitive verb and the preposition (resulting in either a VP or PP), as shown in figures 5 and 6.

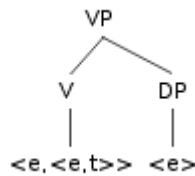


Figure 5: Verb phrase (VP) semantic formation (V taking DP as an argument)

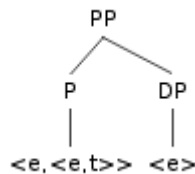


Figure 6: Preposition phrase (PP) semantic formation (P taking DP as an argument)

Radford's Minimalist Syntax (2004) (based on Chomsky's Minimalism) provides for binary trees,

so we use that and assume that parsing (backward or forward) serially is equivalent to moving up/down the tree with no explicit branching required. This required us to change our ATN-based parser, however, which is based on an older sentential non-binary model (figure 7) of tree generation rather than a tense phrase (TP) binary model (figure 8).

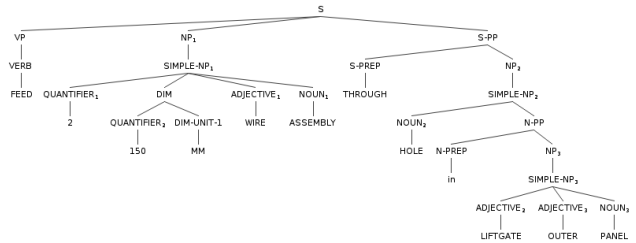


Figure 7: Example of syntax tree for Standard Language sentence

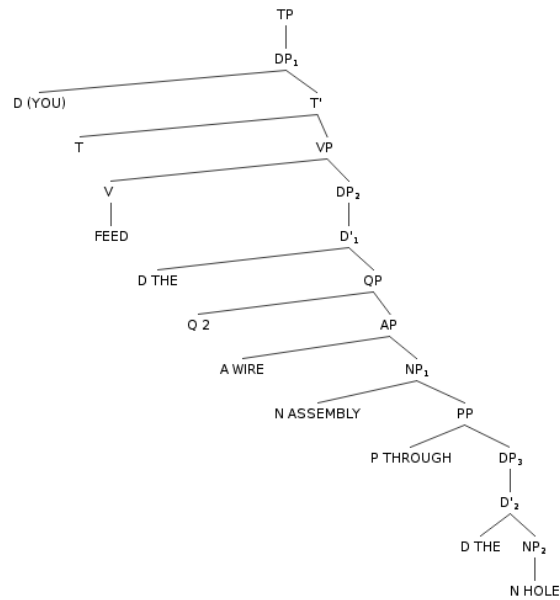


Figure 8: Example of a TP binary syntax tree

We assume that with a verb (V) acting as a semantic function with a determiner phrase (DP) argument (within a Verb Phrase VP) and a preposition (P) acting as a function with a DP argument (within a Prepositional Phrase PP) that the part of speech immediately following the V or the P will be the D (Determiner "the"). We also assume that since the quantifier (Q) takes an argument of type $D_{\langle e,t \rangle}$ that it is taking a noun (N), adjective (A), or intransitive V and therefore will occur higher on the proposed tree template than the adjective phrase (AP). Obviously both the

quantifier phrase (QP) and AP serve to modify the N and here we point out only the reasons for QP occurring before the AP, although English does allow for these to be swapped in many cases. The placement of the PP or VP higher in the tree is simply a function of the Standard Language syntax, but does not affect us if we parse the tree from the bottom to the top.

Our test set includes the following sample sentences with the desired (capitalized) definite article placement:

- apply THE 3 sq inch sealer lubricant to THE fender
- hammer THE hammer with THE hammer
- stand to exit THE vehicle
- align THE 12 sq inch area of THE tape
- align-and-seat THE 13 inch seal to THE strap handle aperture
- grind-finish THE 36 inch fender

There are many combinations to deal with and testing and refining of this particular approach is currently in progress. We have defined several operational facts and rules in order of highest to lowest weight:

1. Define what we call "heavy" (i.e. "listen", "lean", "stand") and "light" (i.e. "walk") intransitives which are verbs that never and almost never take an argument
2. Assume that a prepositional phrase modifier is its own verb when the primary sentence verb is "heavy"
3. Assume that no definite article is needed following a "light" primary verb
4. Assume that complementizers (i.e. "that") act as a sub-sentence head in front of a determiner phrase
5. Assume that temporals (i.e. "second", "year") do not become part of a determiner phrase
6. Assume that idiomatic expressions (i.e. "in"/"into"... "half", "gear", "place") do not become part of a determiner phrase
7. Assume that quantifier phrases do not become determiner phrases
8. Assume that the definite article comes after a verb or preposition (as discussed), but not before an article, preposition, verb, complementizer or quantifier

9. Do not insert the definite article when the object of a "TO" preposition is potentially another verb (i.e. "PULL OBJECT TO STOCK" where "STOCK" is either a verb or a noun, as defined in the ontology)

In our ATN enhanced rules approach we see a significant improvement over the ATN parser rules alone, as shown in figure 9 (the result of testing with 719 random sentences from a representative population). When applying all our facts and rules except #9 (ambiguity) we achieve 99.58% accuracy in definite article placement compared to 72.18% with ATN alone. When we include our ambiguity check we fall to 98.89%, but avoid misplaced definite articles that result in semantically incorrect sentences (i.e. "CRANK THE DOOR HANDLE TO THE CLOSE WINDOW").

Our investigation for handling ambiguous sentences and refining other rules is ongoing and includes looking for context from surrounding instructional sentences using an explicit binding assignment set. Statistical analysis of intra-sentence context did not provide useful guidance as ambiguous verbs show up in a variety of similar sentence formations with different semantics.

	Sentence errors per 719			Accuracy
	Severe	Moderate	Low	
Original ATN	35	159	6	72.18%
ATN with rules 1-8	3	0	0	99.58%
ATN with rules 1-9	0	0	8	98.89%

Severe: Misplaced THE, resulting in incorrect sentence
 Moderate: Missing THE placement before direct object
 Low: Missing THE placement before indirect object

Figure 9: Comparison of ATN and ATN+ result

5.2 Analysis of human error

In order to help understand the challenges our users face in using Standard Language, which facilitates MT by providing better source text, we are conducting a collection and analysis of source text errors. Recent metrics show that 19% of Standard Language submissions must be resubmitted due to various human errors. 14% (of the 19%) are unrelated to Standard Language grammar (such as not assigning a part in the system when the instruction calls for a vehicle part). Upon initial analysis we have formulated a list of common error categories and examples, reduced to nine linguistic failure types defined by

Fromkin (2000), and a tenth type created by us and specific to use of a controlled language. Examples are shown below with percentages of Standard Language error categories from a sample of 20,018 errors:

1. Spelling/typo (32.72%) – author has incorrectly spelled an intended word
2. Theta criterion (5.57%) – violation of required theta-role for agent, patient-experiencer, theme, goal or possessor
 - (you) obtain
 - missing patient theta-role
 - (you) hammer with the hammer
 - missing patient theta-role
3. S-Selection (semantic selection) (0.55%) – incorrect meaning in use of verb and subject or complements
 - (you) hammer the bolt
 - category correct, but semantically incorrect
4. C-Selection (categorical selection) (3.12%) – incorrect category in use of verb and its complements
 - (you) obtain in the vehicle
 - verb expects a determiner, not preposition
5. Linking property (of the verb) (0%) – violation of inherent dependence between verb, arguments and syntactic positions
 - (you) said that operator described the part
 - if meant to mean that "you" described the part
6. Case (0%) – violation of case of corresponding complements
 - (you) obtain you tools
 - implied subject pronoun "you" is always nominative
7. Agreement / Tense (0%)
 - (you) obtained the part
 - standard language is imperative present tense
8. X-Bar (X') (42.77%) – violation of English syntax
 - (you) the part obtain
9. Scope/Ambiguity (QP) (0.81%) – violation of c-command, with potential for ambiguity
 - (you) obtain some part from every rack
 - ambiguous part

10. Vocabulary/Control (14.46%) – violation of controlled language vocabulary

- (you) assuage some part from every rack
 - English verb; not Standard Language verb

The initial categorization of these types of errors will now allow us to assign collected error metrics to meaningful categories that can then be prioritized. Subsequent work may then include online assistance such as spelling and grammar checking, enhancement of Standard Language itself to avoid the potential for certain error types, automated error correction based on learned behaviors and corrections, and additional pre-processing to adjust the source text (without affecting meaning) to allow for appropriate translations.

6 Conclusion

In this paper we have shown some of the challenges in adopting Machine Translation in a large enterprise, including both technical and user expectation factors. With advances in MT such as hybrid (rules-based and statistical) approaches, adoption of web-based technologies, and a service oriented approach we are finding that successful implementation across the enterprise is becoming more feasible and accepted. Engagement of the end-user as part of a “grass roots” approach is critical, as expectations can be measured against practical gains more readily when the systems adoption comes from tackling daily tasks and word of mouth provides assurances of meaningful success.

Acknowledgments

Thank you to AMTA for allowing us to share some of our work, and especially to Mike Dillinger and Ray Flournoy (AMTA 2012 Program Committee) for their review and acceptance of our materials.

References

- Eugene Charniak, Christopher Riesbeck, Drew McDermott, and James Meehan. 1987. Artificial Intelligence Programming, Lawrence Erlbaum Associates, pp. 304-336.
- Victoria Fromkin. 2000. Linguistics: An Introduction to Linguistic Theory, Wiley-Blackwell.
- Willen-Olaf Huijsen. 1998. Controlled Language – An Introduction. Proceedings of the Second International Workshop on Controlled Language Applications (CLAW 98), 1--15.
- Irene Heim and Angelika Kratzer. 1998. Semantics in Generative Grammar. Blackwell Publishing.
- Evgeny Matusov and Selçuk Köprü. 2010. “AppTek’s APT Machine Translation System for IWSLT 2010”, Proceedings of the 7th International Workshop on Spoken Language Translation, pp. 29-36.
- Andrew Radford. 2004. Minimalist Syntax. Cambridge.
- Nestor Rychtycky. 2007. "Machine Translation for Manufacturing: A Case Study at Ford Motor Company" AI Magazine, vol. 28, no. 3, (Fall 2007), pp. 31-44.
- Sylvain Surcin, Elke Lange, and Jean Senellart. 2007. “Rapid development of new language pairs at SYSTRAN”. MT Summit XI, 10-14 September 2007, Copenhagen, Denmark. Proceedings; pp.443-449.