Machine Translation (2011) **25(2)**, pages 87–105 (preprint)
Final paper available at: `http://dx.doi.org/10.1007/s10590-011-9099-4`

Deep Open-Source Machine Translation

Francis Bond · Stephan Oepen · Eric Nichols · Dan Flickinger · Erik Velldal · Petter Haugereid

Received: 15 October 2010 / Accepted: 14 September 2011

Abstract This paper summarizes ongoing efforts to provide software infrastructure (and methodology) for open-source machine translation that combines a deep semantic transfer approach with advanced stochastic models. The resulting infrastructure combines precise grammars for parsing and generation, a semantic-transfer based translation engine and stochastic controllers. We provide both a qualitative and quantitative experience report from instantiating our general architecture for Japanese–English MT using only open-source components, including HPSG-based grammars of English and Japanese.

Keywords Machine Translation · Open Source · Semantic Transfer · HPSG · MRS

1 Introduction

While there have been many advances in the field of machine translation (MT), it is widely acknowledged that current systems do not yet produce satisfactory results, especially for typologically distant language pairs. Contemporary statistical MT systems are both robust and of fair quality, but only for language pairs with similar word order and domains where there is a large amount of existing parallel text. Changing the type of the text to be translated causes the quality to drop off dramatically (Paul 2006). Quality is proportional to the log of the amount of training data (Och 2005), which makes it expensive to continue to improve system quality just by adding more

Francis Bond and Petter Haugereid
Division of Linguistics and Multilingual Studies, Nanyang Technological University, Singapore;
E-mail: bond@ieee.org, {fcbond,petterha}@ntu.edu.sg

Stephan Oepen and Erik Velldal
Department of Informatics, University of Oslo, Norway; E-mail: {oe,erikve}@ifi.uio.no

Eric Nichols
Graduate School of Information Sciences, Tohoku University, Japan;
E-mail: eric@ecei.tohoku.ac.jp

Dan Flickinger
Center for the Study of Language and Information, Stanford University, USA;
E-mail: danf@stanford.edu

data. On the other hand, rule-based systems are still widely used commercially, as they produce relatively constant output and can be tuned to new domains by the use of user dictionaries (Sukehiro et al 2001), although these changes are limited in scope. Therefore, many researchers have come to recognize that no single paradigm solves all of the problems necessary to achieve high coverage while maintaining fluency and accuracy in translation (Way 1999).

In this article we explore an MT approach based on the fundamental position that translation is (largely) a problem of meaning preservation, and thus should be based on an actual analysis and explicit representation of language meaning—an approach to natural language processing (NLP) often dubbed 'deep'. For example, we show that by generalizing on a semantic level one of the most difficult problems in Japanese–English MT—massive differences in word order—can easily be addressed. In general, the more different languages are, the more useful deep analysis becomes. For example, in the Spanish open-source initiative OpenTrad two engines were developed: Apertium, a shallow-transfer engine, is used to translate between Spanish and Catalan, Galician, and Portuguese (Forcada et al 2011), while Matxin, a structural transfer system, is used only to translate Spanish into Basque, a very different language typologically (Mayor et al 2011).

Much current research on MT involves the addition of more syntactic information, as exemplified by the recent series of workshops on Syntax for Statistical Machine Translation. However, while the syntactic level doubtless is useful, we still consider semantics to be the appropriate 'ultimate' level of cross-lingual representation—this is in part because more work can be delegated to reusable, mono-lingual resources.

The long-term aim of our research is two-fold, viz. (a) to develop and release a state-of-the-art infrastructure for MT based on the combination of semantic transfer and stochastic (re-)ranking; and (b) to produce a high-quality, easily extensible, and robust Japanese–English instantiation of this paradigm. Our approach here is to concentrate on translation quality and extensibility, without equal concern for coverage. We are able to do this because of the (open-source) availability of robust statistical machine translation (SMT; e.g. Koehn et al 2007): in our view, the primary task for our deep system then is to produce high-quality translations for those inputs it can translate, whereas we will fall back to SMT for all remaining inputs.

This leaves the problem of how to build a system that is both high quality and easily extensible. To gain high quality, we accept the brittleness of a rule-based semantic transfer system. In particular, by using a deep grammar in analysis, we can abstract away from phenomena specific to the source language—such as word order. Likewise, by using a precise grammar in target language generation we ensure that our outputs are grammatical. We then concentrate our manual rule-building efforts on closed-class words, multi-word expressions, and other cases of transfer-level translational divergences. To make the system (relatively) easily extensible, we construct a default set of transfer rule instances from aligned corpora and bilingual dictionaries.

This paper is organized as follows. In Section 2, we introduce the reusable computational grammars and other linguistic resources which make this research possible. In Section 3 we outline the core system architecture and transfer formalism. In Section 4 we describe the Japanese–English MT system (JaEn) in more detail, including the transfer rule acquisition and translation ranking. We conduct quantitative and qualitative evaluation of the system in Section 5. We discuss these results in Section 6 and then briefly outline some future work in Section 7. Finally, we conclude this paper in Section 8.

2 Reusable Computational Grammars and Linguistic Resources

MT based on semantic transfer presupposes monolingual analysis and generation. Our efforts build on a wealth of earlier work in computational grammar engineering, where a combination of (a) widely accepted linguistic theories, (b) stable and declarative formalisms, (c) advances in engineering and evaluation methodology and tools, and (d) sustained, incremental development over the past two decades has enabled the creation of comprehensive, precise,[1] general-purpose grammars for a handful of languages. Specifically, we draw heavily on the open-source repository of the Deep Linguistic Processing with HPSG Initiative (DELPH-IN)[2], which provides us with, among other things, large-scale bi-directional grammars (which both parse and generate) for English (the English Resource Grammar: ERG; Flickinger 2000) and Japanese (JACY; Siegel and Bender 2002)[3], as well as with software for parsing (LKB and PET; Copestake 2002; Callmeier 2002), generation (Carroll and Oepen 2005), and stochastic experimentation and regression testing ([incr tsdb()]; Oepen and Flickinger 1998). Developing and maintaining multi-lingual resources at this scale would be prohibitively expensive for any individual player (all but the largest corporations, perhaps), and our current research on MT would have been impossible without access to this wealth of prior work. As a fallback translation engine and a source of translation probabilities we use Moses (Koehn et al 2007), an open-source toolkit for phrase-based statistical machine translation.

We also make heavy use of open linguistic resources, especially JMDict, a Japanese → Multilingual dictionary (Breen 2004). JMDict consists of Japanese index words with one or more English translations. It has approximately 150,000 main entries, with an additional 350,000 proper names. The dictionary is primarily used by non-native speakers of Japanese as an aid to read Japanese. It has many contributors, and is increasing in size at the rate of almost 1,000 entries a month (Bond and Breen 2007). For parallel text, we primarily use the Tanaka Corpus of Japanese–English sentence pairs (Tanaka 2001), currently maintained by the Tatoeba Project.[4]

We list the major components and their licenses in Table 1. The core system components (marked with an asterisk) are all made available from the LOGON repository.[5] All the parts of a working Japanese–English MT system can be downloaded in a single command, and it runs out of the box on most current Linux distributions.

In many respects similar to DELPH-IN is the Parallel Grammar (ParGram) network (grounded in the LFG rather than the HPSG theory of grammar), which is the provider of the Norwegian analysis component in the Norwegian–English MT system (NoEn). Although the Norwegian grammar itself (Dyvik 1999) has been open-sourced, ParGram resources require the proprietary XLE software (which is available for non-commercial use upon request from the Palo Alto Research Center) to run. Therefore, in this article, we focus on the fully open-source Japanese–English instantiation of our infrastructure,

[1] A precise grammar distinguishes between grammatical and ungrammatical sentences.

[2] `http://www.delph-in.net`

[3] Both grammars were originally developed within the Verb*mobil* MT effort, but over the past decade have been used for a wide variety of tasks, including automatic email response, ontology learning, and grammar checking. The grammars are being continuously developed by separate groups of researchers, but like all DELPH-IN resources commit to a uniform format for meaning representation, Minimal Recursion Semantics (see § 3).

[4] `http://tatoeba.org`

[5] `http://wiki.delph-in.net/moin/LogonTop`

Component	Description	License
LOGON *	Hybrid Machine Translation Platform	LGPL
LKB *	Grammar Development Environment	MIT
PET *	Unification-Based Chart Parser	LGPL
ERG *	English Resource Grammar (HPSG)	MIT
JACY *	Implemented Grammar of Japanese (HPSG)	MIT
ChaSen*	Japanese Morphological Analysis	BSD
JaEn *	Japanese–English Transfer Grammar	MIT
JMDict	Japanese–English Lexicon	CC-by-sa
Tanaka Corpus*	Japanese–English Corpus	CC-by-sa
Moses	Statistical Machine Translation Platform	LGPL

Table 1 Major components and choices of open-source licenses. Asterisks indicate JaEn core system components.

although we consider it a strong point of our approach that the level of semantic analysis applied enables genuine cross-framework integration: in the Norwegian–English system, the LFG source language grammar is paired with the same HPSG target language generator as in the Japanese–English system (the ERG).

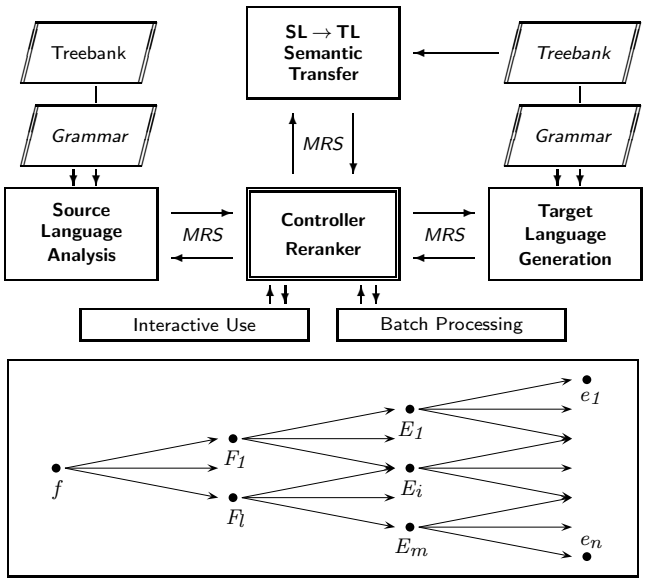
We distribute the version of the Tanaka corpus we use along with the Japanese grammar JACY. It has 147,190 sentence pairs, which we divided into 100 sections of roughly 1,500 sentences each, randomly ordered. We use sections 0–2 as development data, 3–5 as test data and the rest as training data.

3 The LOGON Transfer-Based MT Architecture

Our open-source infrastructure is named after the LOGON project (which originally developed a Norwegian–English MT system). The Norwegian LOGON consortium was made up of the universities of Oslo, Bergen, and Trondheim, with collaborating partners in Germany, Japan, and the US (Oepen et al 2004a, 2007). Between 2003 and 2007, the project expended around fifteen person years of work, in part on building the core infrastructure (which we fully reuse for Japanese–English MT), in part on adapting the Norwegian and English grammars and the Norwegian–English transfer grammar.

LOGON adopts the widely used framework of Minimal Recursion Semantics (MRS; Copestake et al 2005) for meaning representation (see Section 4 below for examples), and MRS serves as the framework-independent interface format among components. Figure 1 shows a schematic view of the system architecture, combining source language parsing, semantic transfer by MRS rewriting, and grammar-based target language generation in a non-deterministic fan-out architecture (with support for parallelization and distribution across a commodity cluster). The role of rule-based components in LOGON is to delineate the space of grammatically and semantically coherent translations, while the ranking of competing hypotheses and ultimately the selection of the best candidate(s) is viewed as a probabilistic task.

Parsing, transfer, and generation each produce, on average, hundreds or thousands of candidate outputs for one input. Hence, exhausting the complete fan-out combinatorics can be prohibitively expensive, and we typically limit the number of hypotheses passed downstream to a relatively small n-best list. Irrespective of the inner workings of individual components, the architecture assumes a stochastic ranking of local hypotheses at each stage—probabilities $P(F_j|f)$, $P(E_i|F_j, f)$, and $P(e_k|E_i, F_j, f)$ in the notation of the fan-out tree from Figure 1. At the same time, it is quite common



Where f and e are the source and target text, and F and E the corresponding MRS.

Fig. 1 Schematic system architecture (top) and resulting fan-out tree (bottom).

for distinct fan-out paths to arrive at equivalent outputs, for example where the same modifier attachment ambiguity may be present in the source and target language. Once fan-out completes, end-to-end reranking is applied on the resulting n-best list, aiming to directly maximize the posterior translation probability $P(e|f)$ in a log-linear model; see below for details. Our linguistic resources, search algorithms, and statistical models draw from contemporary, state-of-the art techniques and ongoing research in larger, non-MT communities. In this regard, the LOGON architecture provides a novel blending of approaches, where the majority of its component parts and linguistic resources have independent value (and are also used in other research efforts and applications).

The LOGON transfer engine—ordered, unification-based, resource-sensitive rewriting of MRS formulae—constitutes a new generic tool which has been adopted for other language pairs and also for non-MT tasks such as paraphrasing. Abstractly, a transfer grammar is composed of a sequence of MRS rewrite rules (MTRs). Transfer rules replace MRS fragments in a step-wise manner. The general form of one MTR is shown schematically in (1), as a four-tuple with components $\mathcal{C}$ (context), $\mathcal{I}$ (input), $\mathcal{F}$ (filter), and $\mathcal{O}$ (output).

(1) $[\,\mathcal{C}:]\;\mathcal{I}\;[\,!\,\mathcal{F}\,] \rightarrow \mathcal{O}$

Here, each of the four components is a partial MRS. Left-hand side components are unified against an input MRS M and, when successful, trigger the rule application; elements of M matched by $\mathcal{I}$ are replaced with the $\mathcal{O}$ component, respecting all variable bindings established during unification. The optional $\mathcal{C}$ and $\mathcal{F}$ components serve to condition rule application (on the presence or absence of specific aspects of M), establishing bindings for $\mathcal{O}$ processing, but do *not* consume elements of M.

Transfer rules are applied in an ordered, breadth-first rewrite process until a fixpoint is reached, where alternate rules for compatible inputs give rise to non-determinism.

A novel feature of the LOGON transfer formalism is its use of typed feature structures and multiple inheritance in the description of MTRs, which facilitates underspecification and modular design. In the following, however, we will mostly present transfer rule examples in their fully expanded form, effectively glossing over the use of typing in LOGON transfer grammars. The typing was essential in creating the Transfer Matrix: a collection of generic transfer correspondence types (organized in an inheritance hierarchy that facilitates further refinement, as needed) which serves as a starter kit for new language-specific transfer grammars. This was shared between NoEn and JaEn, and helped in the rapid development of the second system. As illustrated in Section 4.1 below, large numbers of transfer rules can be (semi-)automatically derived by populating the generic types with actual lexical correspondences, e.g. from a bilingual dictionary or aligned corpora.

Finally, LOGON provides the complete tool chain required to train, evaluate, and apply stochastic models to rank hypotheses at each processing level and end-to-end (including language models, discriminative syntactic and semantic models, SMT models, and others). The system is not a production system — it is designed for flexible experimentation and is both slow (sentences may take over a minute to translate) and memory intensive (on the order of 8 GB).

4 Japanese–English: Architecture and Components

For Japanese–English MT, we populate the LOGON architecture with the DELPH-IN grammars JACY (source language parsing) and ERG (target language generation). The system has been described previously in Bond et al (2005) and Nichols et al (2007). The Japanese–English transfer grammar (JaEn) is being developed on the basis of the LOGON Transfer Matrix, which we populate through a combination of manual work and automatic rule acquisition (see below). We combine LOGON with Moses (Koehn et al 2007), both to provide a back-off system for improved robustness as well as to derive alignment information for transfer rule construction (and of course also as a general point of comparison). We learn lexical transfer correspondences from parallel corpora and the Japanese – English dictionary JMDict (Breen 2004), and develop primarily against the bilingual Tanaka Corpus (Tanaka 2001).

Before describing how we compiled the transfer grammar, we give an example of the MRS meaning representation and the transfer process. Consider the Japanese sentence (2) and its (moderately simplified) semantic representation (3):

(2) その うそ は 子供 たち が　　ついた
sono uso-wa kodomo-tachi-ga　　*tsui-ta*
that lie-TOP child-and others-NOM breathe out-PAST
The children told that lie.

(3) $\langle h_1,$
$\left| \begin{array}{l} h_3{:}\_\mathsf{sono\_q}(x_5,\ h_6,\ h_4), h_7{:}\_\mathsf{uso\_n\_2}(x_5), h_7{:}\_\mathsf{wa\_d}(e_9,\ e_8,\ x_5), \\ h_{12}{:}\mathsf{udef}(x_{11},\ h_{14},\ h_{13}), h_{10}{:}\_\mathsf{kodomo\_n}(x_{11}), h_{10}{:}\_\mathsf{tachi\_a\_pl}(u_{15},\ x_{11}), \\ h_{16}{:}\_\mathsf{tsuku\_v\_6}(e_2\{\mathsf{TENSE}\ \mathit{past}, \mathsf{SF}\ \mathit{prop}\},\ x_{11},\ x_5) \end{array} \right|$
$\{\ h_{14} =_q h_{10},\ h_6 =_q h_7\ \}\ \rangle$

The MRS in (3) can be read as a 'flat' underspecified logical form, whose body consists of *elementary predicates* (EPs), such as $h_7{:}\_\mathsf{uso\_n\_2}(x_5)$ ("lie") or $h_{16}{:}\_\mathsf{tsuku\_v\_6}(e_2, x_{11}, x_5)$ ("breathe out"). These correspond to atomic formulas in predicate logic,

where logical variables like the entity x_5 or the attaching event e_2 can establish links across EPs. All elementary predicates have a unique distinguished argument (ARG0) which can be used to identify the predicate, and many also have arguments (labeled with ARG1–ARG3) whose interpretation depends on the predicate (Copestake 2009).

Generalized quantifiers (e.g. _sono_q, "that") and other operators (e.g. modals or scopal adverbs) establish scopal relations, which are captured in terms of *handles* (the h_i prefixed to each EP) and (potentially underspecified) handle constraints (the $h_i =_q h_j$ near the end of (3)). In the following, we will ignore details of scope; see Copestake et al (2005) for details.

MRS abstracts away from the surface syntactic structure in several important ways, although it is far from being an interlingua. For example in (3), the noun phrase *sono uso* "that lie" is recognized as the object of the verb *tsuku* "breathe out", even though it comes before the subject, and is marked with a topic marker *wa*, rather than the accusative marker. Also, in the noun phrase *kodomo-tachi*, the noun *kodomo* "child" is recognized as the semantic head of the phrase, and the suffix *-tachi* "and others" is treated as a modifier.

For this running example, the expected transfer output is shown in (4):

(4)
$$\langle\, h_1,\ \left|\begin{array}{l} h_3{:}\mathsf{def\_udef\_a\_q}(x_5,\ h_6,\ h_4),\ h_7{:}\mathsf{\_child\_n\_1}(x_5\{\mathsf{NUM}\ \mathit{pl}\}),\\ h_8{:}\mathsf{\_tell\_v\_1}(e_2\{\mathsf{SF}\ \mathit{prop},\ \mathsf{TENSE}\ \mathit{past}\},\ x_5,\ x_9),\\ h_{10}{:}\mathsf{\_that\_q\_dem}(x_9,\ h_{12},\ h_{11}),\ h_{13}{:}\mathsf{\_lie\_n\_1}(x_9\{\mathsf{NUM}\ \mathit{sg}\}) \end{array}\right|$$
$$\{\, h_{12} =_q h_{13},\ h_6 =_q h_7 \,\}\, \rangle$$

Although the resulting syntax of the generated English sentence will be quite different, the topology of the MRS is similar. There are three main differences. The first, and most obvious, is that the predicate names are different: simple lexical transfer rules transform, for example _uso_n_2 into _lie_n_1. A slightly more complicated rule is used to translate _tsuku_v_6, conditioning on (the class of) its deep object (the ARG2): If the object matches _uso_n then _tell_v_1 is selected as the translation of _tsuku_v_6; otherwise, the default translation would be _breathe_v_out.

Most transfer rules are semi-automatically compiled from parallel corpora and a bilingual dictionary, as described below (§ 4.1). The order of predicates in the Japanese and English MRSs is also different, but this is unimportant; the set of EPs forms an *unordered* bag. The flat, order-insensitive structure makes it easy to apply transfer rules to MRSs: a partial match can be established for any subset of EPs, as long as structural relations (shared variables and handles) can be unified.

As a second, more interesting difference, in (4) the Japanese suffix *-tachi* "and others" has been consumed in a rule that marks the logical variable representing the noun phrase that it used to modify (i.e. the ARG0 of _child_n_1) for plurality. Finally, the covert Japanese quantifier udef_q has been replaced by an underspecified quantifier def_udef_a_q that subsumes the (semantics of) the English articles *a*, *the*, or no article. This leaves the choice of article to the target language generator (and its stochastic ranking of competing English realizations). The topic marker is ignored here (and deleted in transfer), although ideally it should be used to influence the choice of article and possibly trigger topicalization in English.

The actual English translations for our example are shown in (5). Because *child* is constrained to be plural, the only possible determiners are *the* or no article, so these two possibilities are generated.

(5) a. The children told that lie.

b. Children told that lie.

Of course, semantic transfer does not solve all the problems of machine translation. In particular, problems of lexical choice remain (e.g. should うそ *uso* be translated as "lie" or "falsehood"), as well as other fundamental problems due to substantive typological differences between languages. For example, Japanese does not distinguish between singular and plural, or countable and uncountable, so most noun phrases are very underspecified. We currently approach these problems by non-deterministically producing multiple candidates and selecting among them using stochastic models, as described in Section 4.2.

We adopt MRS-based transfer because it is easy to generalize over structures. This eases translation from Japanese into English in several ways. Example (6) shows how questions can be difficult for SMT systems to translate due to their unusual word order. This sentence is not a problem for JaEn: both arguments of the verb 作る *tsukuru* "make" are embedded in its argument structure and transferred to the English grammar intact. In the actual sentence, *tsukuru* is combined with the politeness marker in Japanese: *mashita*, inflected for past. The Japanese parser gives the same semantic representation to the polite form *tsukurimashita* as the standard form *tsukuru*, allowing general translation rules to work without change.

(6) 何 を 作り まし た か 。
nani-wo tsukuri-mashi-ta ka .
what-ACC make-POLITE-PAST QUES .

What did you make? (Reference)

What did you make? (JaEn)

What made you? (Moses)

Our translation system has similarities with other systems like ALT-J/E (Ikehara et al 1996) or the Meaning-Text Theory approach to machine translation presented in Mel'čuk and Wanner (2006). According to Mel'čuk and Wanner, the transfer is carried out at the *Deep-Syntactic Structure*, which abstracts away from surface-syntactic structure. Language specific phenomena like restricted lexical co-occurrence and language-specific constructions are treated in analysis/parsing or synthesis/generation. This eases the burden on the transfer component. While the Japanese and English grammars used for parsing and generation in our project assign single predicates to multi-word expressions like *in front of*, *look up*, and *by and large*, the Meaning-Text Theory goes further, and assumes a decomposition of surface expressions into keywords and lexical functions. In (2), which glosses as "Children breathed out that lie", "breath out a lie" will get the deep syntactic structure 'Oper-1(LIE)', where 'Oper-1' is a lexical function (loosely meaning "do") applying to the keyword LIE. This deep syntactic structure can then be directly transferred. The advantage of the Meaning-Text Theory approach is that once the lexical resources are in place, the effort to build transfer grammars between language pairs is greatly reduced. However, the main challenge is to build up the lexical resources, and so far the approach does not have a large-scale implementation.

4.1 The Transfer Grammar

Example (7) shows the most basic form of transfer rules, as used in our example above:

(7) $\langle \_, \{ \boxed{h_0}\text{:\_uso\_n\_2}(\boxed{x_0}) \}, \{ \} \rangle \rightarrow \langle \_, \{ \boxed{h_0}\text{:\_lie\_n\_1}(\boxed{x_0}) \}, \{ \} \rangle$

This rule rewrites just the predicate symbol (formally, it seeks to match an MRS fragment comprised of a single EP), while the 'boxed' variables indicate that both the handle and ARG0 variable are preserved between the input ($\mathcal{I}$) and output ($\mathcal{O}$) of the rule. In this kind of simple transfer correspondence anything but the lexical predicates is shared across large numbers of transfer rules. Such generic aspects of transfer rules are captured in the type hierarchy, and in practice the specification of the above rule is simplified to the following:

(8) NOUN_MTR := $\langle \_, \{ \boxed{h_0}\text{:}\_(\boxed{x_0}) \}, \{ \} \rangle \rightarrow \langle \_, \{ \boxed{h_0}\text{:}\_(\boxed{x_0}) \}, \{ \} \rangle$

(9) NOUN_MTR $\wedge$ { _uso_n_2 } $\rightarrow$ { _lie_n_1 }

Here, the correspondence type NOUN_MTR captures the equalities between $\mathcal{I}$ and $\mathcal{O}$ that hold independently of the actual predicate symbols. We were able to use many correspondence types from the LOGON Transfer Matrix (which was originally developed when we started deriving the Japanese–English system on the basis of the existing NoEn transfer grammar). JaEn inherits from LOGON types for open-category lexical items such as common nouns, adjectives, and verbs with a variety of (regular) argument structure configurations. In addition, the LOGON Transfer Matrix contains a number of correspondence types to specify rules for quantifiers, particles, or conjunctions, providing much of the framework needed to develop JaEn. A number of smaller MT systems built on the LOGON infrastructure also share these rule types.

Closed-class rules We hand-write rules for closed-class lexical items, such as quantifiers, pronouns (including zero pronouns), prepositions, copulas and auxiliaries. We also have detailed rule sets for handling temporal expressions, tense, aspect and other phenomena that often differ radically between languages. There are roughly 2,000 of these rules.

Open-class rules The majority of transfer rules are lexical rules we acquire from either JMDict or aligned translation corpora. We use rule templates to generate transfer rules from input source and target MRS structures (similar in spirit to Nygaard et al 2006). Template mappings are used to identify translation pairs where there is a compatible rule type that can create a transfer rule. For example, the template mapping: `T([noun], [adjective, noun], n-adj+n)` matches the JMDict entry "悪玉 /bad character/" and creates a rule to translate a Japanese noun into an English adjective-noun sequence: `mtr` = ⟨悪玉 *akudama* → bad character⟩.

We are using Moses (Koehn et al 2007) and Anymalign (Lardilleux and Lepage 2009) to generate phrase tables from a collection of four Japanese–English parallel corpora and one bilingual dictionary. The corpora are the Tanaka Corpus (2,930,132 words: Tanaka (2001)), the Japanese Wordnet Corpus (3,355,984 words: Bond et al (2010)), the Japanese Wikipedia corpus (7,949,605),[6] and the Kyoto University Text Corpus with NICT translations (1,976,308 words: Uchimoto et al (2004)). The dictionary is Edict, a Japanese–English dictionary (3,822,642 words: Breen (2004)). The word totals include both English and Japanese words.

[6] The Japanese–English Bilingual Corpus of Wikipedia's Kyoto Articles: `http://alaginrc.nict.go.jp/WikiCorpus/index_E.html`

We acquire some 105,000 rules automatically. Because we continue to add to the parallel corpus and JMDict is constantly being updated and extended, we keep the automatically acquired rules separate from the hand-crafted rules, and periodically update them. So as not to overwhelm the translation system, we only keep automatically acquired rules with a translation probability above a certain threshold (the exact threshold varies from rule to rule). The rule construction is described in more detail in Haugereid and Bond (2011).

Transfer rules are ordered so that transfer words that cover multiple source words are applied first, and then for multiple translations for a single word, they are ordered according to the target language relative frequency. We have recently started learning rules from semi-translated sentences — if JaEn can translate all but a handful of predicates, then we try to build a rule for them by matching with the parse of the reference translation. This is particularly useful for learning rules that include derivation: JMDict has "日本 /Japan/" but not "日本 の /Japanese (lit. of Japan)/". The Japanese side is entirely productive, and the English is predictable by a native speaker but we need a rule for this in the translation. This is learned directly from the parallel corpus.

Because frequent words are often idiomatic in their usage, we are in the process of expanding the core open-class translations by hand — ideally we would like to verify the most common word pairs, especially those with different argument structure.

4.2 Ranking Translations

JaEn employs a combination of five different stochastic models to rank intermediate hypotheses and candidate translations. The first such model is applied off-line: when creating transfer rules from dictionaries, rules are ordered according to the phrase table probabilities. The ordering of alternate transfer rules for equivalent input fragments (e.g. the variants of translating うそ *uso* as either "lie" or "falsehood") matters where an exhaustive breadth-first rewrite process would be intractable. To reduce exponential growth in the number of candidate transfer results, we impose a ceiling of three rules to be explored at each point. Increasing this improves accuracy, at a cost of greatly reduced speed.

In the actual run-time system, stochastic models are used in each phase of parsing, ranking, and generation. A fan-out ceiling of five is imposed on each step, so only the five top-ranked source language analyses are passed on to transfer, which in turn will send a maximum of 25 MRSs to generation (five per branch).[7] Finally, the top five target language realizations from each branch are gathered together and the results are reranked (across branches) using a global discriminative model. The following paragraphs provide more background on the individual models.

Parse and Realization Ranking Parse ranking is done using a model trained on 7,000 manually treebanked sentences from the training set of the Tanaka Corpus (Bond et al 2008). The model is a discriminative log-linear model, which uses structural features from the parse derivations such as local sub-tree configurations, dominance relations, and n-grams of lexical types—adapting the LinGO Redwoods approach used with the ERG to Japanese (Fujita et al 2007; Oepen et al 2004b). A similar model is used for ranking the generator output (realization ranking: Velldal and Oepen 2006), although

[7] This number was chosen based on experience with the LOGON NoEn system.

with the addition of an n-gram language model as an auxiliary distribution. Training data is automatically obtained from the English LOGON treebank, and the language model is trained on the British National Corpus (BNC; Burnard 2000).

Because Japanese is so different from English, we build a realization model based on underspecified input. In particular, we underspecify number, definiteness and person. A sentence like *Children told a lie* would thus become tell(child,lie) with {*sg,pl*} generalized to *num* and the determiner replaced with a supertype. From this we generate *The child told a lie; A child told lies; Children told the lie; ...*. The realization ranker can thus learn defaults: e.g. *lie* is usually singular (in our corpus), and so on. In a case like (5), where the input is slightly specified, the English generator only generates the possible translations and the model has fewer things to rank. This kind of generation ambiguity was not present in the original Norwegian–English, where the realization ranker was trained on sets of paraphrases generated from the MRS associated to the top-ranked analysis in the English treebank.

Transfer Ranking To rank transfer outputs, we trained a model on the training section of the Tanaka corpus. It ranks English MRSs using a reduction into variable-free Elementary Dependency triples (Oepen and Lønning 2006); semantic triples are then represented in normalized textual form and ranked according to the perplexity scores of a smoothed tri-gram language model trained on the MRSs from the first best parse of some 15,000 English sentences.

End-to-End Reranking Again, we use existing technology from the LOGON project, reranking candidate translations on the basis of a log-linear combination of features reflecting the derivation of each candidate. This is motivated because (a) the component-internal stochastic models at each stage are fallible and (b) besides analysis-, transfer-, and realization-internal information, there are additional properties of each hypothesized pair $\langle f, e \rangle$ that can be brought to bear in choosing the 'best' translation. These can include, for example, a measure of how much reordering has occurred among corresponding elements in the source and target language, or the degree of harmony between the string lengths of the source and target. Although very different in practice, the overall philosophy is similar to that described by Och and Ney (2002) for SMT.

Oepen et al (2007) discuss eleven features directly available in the LOGON infrastructure. After testing various combinations, we found that for JaEn the (unnormalized) scores from the parsing, transfer, and generation models, a separate n-gram language model again trained on the British National Corpus and lexical correspondence probabilities (obtained from the Tanaka Corpus using GIZA++) all helped. Unlike in the Norwegian–English experience, we have yet to obtain tangible improvements from including a distortion model.

5 Evaluation

In this section, we evaluate our Japanese–English system by analyzing its coverage over the Tanaka Corpus test sections. The system tested was the result of around twenty four months of development targeting the Tanaka Corpus, during which we increased our end-to-end coverage on the development data from 8.5% to 26.1% while improving the translation quality. Full statistics for parsing, transfer, and generation coverage are given in Table 2.

Table 2 Coverage on the Tanaka Corpus.

Data	Parsing	Transfer	Generation	End-to-End
Tanaka Test (003–005)	80.31%	60.1%	54.1%	26.1%

We can currently parse 80% of the entries in the Tanaka Corpus' development and test sets. Our transfer coverage is 60%, and that of generation is 54%. The end-to-end coverage is the product of these: 26%.

In comparison to commercial rule-based systems, or a statistical machine translation system, the coverage is very low. Nevertheless, we have constructed a firm foundation for future research, with quite limited resources (around five person-years of development on JaEn). The main bottleneck is the lack of transfer rules. Generation coverage on well-formed input is close to 100%; the coverage is low here because the transfer rules are not producing appropriate English semantic representations. We can increase transfer coverage significantly by transliterating unknown words, but this decreases the accuracy, so we chose not to.

5.1 Automatic Evaluation

We compared our hybrid system to Moses (Koehn et al 2007), using two automatic evaluation methods: BLEU and METEOR.[8] Moses was trained on the Tanaka Corpus using a non-factored phrase-based model (the same system used in Nichols et al (2010)). BLEU scores (Papineni et al 2002) were calculated using the `multi-bleu.perl` implementation distributed with Moses. METEOR (Banerjee and Lavie 2005) was calculated with the NIST software, enabling stemming and WordNet synonym matching (version 0.6 with the flag "–modules exact wn_stem wn_synonymy").

The evaluation results are given in Table 3. We conducted evaluation using standard, written English with capitalization and punctuation.

Table 3 Comparison of JaEn and Moses.

System	BLEU	METEOR	Human
JaEn	10.07	38.51	**52.75**
Moses	**23.85**	**51.65**	47.25

BLEU and METEOR are for all sentences translated by both systems.
Human is for a sample of 100 sentences.

Our system is outperformed by Moses using these automatic metrics. The major reason is that Moses is optimized for BLEU while JaEn is not. In particular, because we learn many translations from a lexicon, not the corpus, we often end up with out-of-domain translations that—while acceptable—will not match strict n-gram based metrics. METEOR, which has a loser match criterion, with some allowances for similar words, shows the systems to be much closer.

[8] We also measured NIST and NEVA, with similar results.

5.2 Human Evaluation

Because n-gram based automatic metrics are known to favor systems that use n-gram models, we also conducted a small-scale human evaluation of the quality of JaEn and Moses. 100 sentences were randomly selected from the cross-section of the Tanaka corpus test data that both JaEn and Moses could translate.

There were two evaluators: one was a native Korean speaker, fluent in English and Japanese, with experience in Japanese–English translation, and the other a Japanese linguistics exchange student. They were each shown the Japanese source sentence, its English reference translation, and the output from the two systems labeled "System A" and "System B." The labels were randomly selected for each sentence; the evaluator did not know which system produced a given output until the evaluation was concluded. The evaluator selected the preferred system for each translation (which gets it a score of 1.0), with the option of ranking them equally (in which case each gets a score of 0.5), a reliable and cost-effective method. In this evaluation, JaEn was preferred to MOSES by both evaluators, with scores of 53 and 52.5 respectively.

According to the evaluators, who saw the results for each system revealed after the evaluation, Moses did better in choosing the content words, but was often ungrammatical. Further, Moses would occasionally lose an essential element (such as a negation). Both systems dealt badly with pronouns (which are typically omitted in Japanese). Overall, neither system gave translations of sufficiently high quality that an English speaker with no knowledge of Japanese would always reliably understand the intended meaning.

5.3 Qualitative Evaluation

We give a qualitative comparison of the two systems in (10–12). This small selection of sample translations illustrates the strengths and weaknesses of each of the systems.

(10) 私はいやいやその仕事をした。
"I did the work against my will." (Reference)
"I did that work unwillingly." (JaEn)
"I did the work against his will." (Moses)

(11) リストに彼女の名前がなかった。
"Her name didn't appear on the list." (Correct)
"His name didn't appear on the list." (Reference)
"There was not any name of hers on the list." (JaEn)
"Her name on the list." (Moses)

(10)–(11) are typical examples where our system translates more accurately than Moses. In (10) Moses produces fluent idiomatic output, but the meaning is not preserved—it was the speaker who didn't want to do the work, not some third person. JaEn avoids using pronouns when it can't get the referent correct. An alternative would have been to co-index the pronoun with the subject (and thus produce *X did Y against X's will*). (11) shows both an error in the reference translation (*his* where the source means *hers*) and Moses losing the negation. JaEn's translation could be made more fluent, but at least preserves the meaning.

(12) メイド は テーブル に ナイフ と フォーク を 並べ た 。
"The maid arranged the knives and forks on the table." (Reference)
"The maid enumerated the knife and hawk on the table." (JaEn)
"The maid on the table arranged the knives and forks." (Moses)

(12) shows a typical example where JaEn does not do so well—the meaning can be guessed, but the lexical choice is poor: *hawk* instead of *fork* (the result of our misparse of JMDict) and *enumerate* instead of *arrange*.

6 Discussion

Semantic transfer machine translation projects have a long history. However, the majority of these systems are of a proprietary nature; when the project concludes, the resources that were developed are rarely made available to other researchers, so it is difficult for the field to directly benefit from what was learned throughout the course of development. A notable exception to this is the OpenLogos System (Barreiro et al 2011), who have opened up their system for collaborative development.

We recognize the problem that closed resources pose to machine translation systems that use detailed linguistic representations, so one of our goals is to make our resources freely available to other researchers. Every component of our machine translation system, from the parser to the grammars, is available as open source, including the transfer rules and test corpora.

Further, an important part of producing JaEn has been contributing back to the open-source projects that made this work possible. Our main contribution is the machine translation system itself, but we have also done: development work on the source and target language grammars and LOGON system; amendments and additions to JMDict and the Tanaka Corpus; packaging Moses and LOGON for easy installation on Ubuntu Linux systems[9] and producing prototype systems for Korean–Japanese, Spanish–Japanese, and Norwegian–Japanese. We believe that this kind of feedback between systems is an important part of improving the base level of NLP research. Of course, we have also benefited from improvements in the various components, especially from work on greater robustness in parsing and generation.

JaEn is also being successfully used in teaching machine translation and grammar engineering. The system allows all the intermediate steps to be seen and provides a platform for student experimentation.

The system as it currently stands is an interesting research system, but not a useful production system. However, it has the potential to improve its translation quality, at least for those sentences it can translate. Current work is focusing on improving quality, rather than coverage — we can always fall back to the more robust SMT system for sentences we cannot handle. This means an emphasis not only on acquiring knowledge, but also on making sure that new and existing rules maintain or improve quality. We are especially looking at adding rules with more context (multi-word to multi-word) both learned from JMDict and parallel corpora.

Our experience in the machine translation industry in Japan led us to believe that dictionary maintenance is an unsolved problem for commercial rule-based systems, leading to the "knowledge bottleneck". Once a product has been released, there is

[9] Packages are available from Ubuntu NLP: `http://cl.naist.jp/~eric-n/ubuntu-nlp/`

little incentive to keep improving the dictionaries, and experienced developers typically move on to other projects. Because of this, we have tried to cooperate with existing lexical projects such as JMDict, rather than producing our own lexicon, which would have to be maintained separately. One problem with this is that, because the end users of JMDict are people, the translations are often more informative than the most common translation equivalents. For example, 医者 *isha* "doctor" is translated as *medical doctor*, and フランス語 *furansugo* "French" as *French language*, in order to disambiguate them from *Doctor [of Philosophy]* and *French [national]* respectively. These are both correct translations, but they are not ideal for an MT system: in context, the meaning is normally clear and a translation of just "doctor" or "French" would be preferable. We are working with JMDict to add these extra common translations, which also make the dictionary more useful for English–Japanese glossing. We also try to add new word pairs to JMDict and rebuild our lexicon from there, rather than adding them to our local transfer rules.

7 Future Work

In addition to the constant work on improving the quality of the system by expanding the inventory of rules, and providing feedback to the component grammars, we are working on learning more and better transfer rules. We have two approaches. The first is to improve the current extraction by preparing more and better templates. The second is to learn transfer rules from parsed parallel text: we parse both the source and target language sentences, then transfer the source and attempt to align the (possibly partial) translation with the parse of the reference translation. Aligned MRS structures can then be rewritten as rules. A similar approach has been taken by Jellinghaus (2007). The main differences are that they only align very similar sentences, always start the alignment from the root (the handle of the MRS), and directly align the source and target MRSs without partial translation. This is another approach to avoiding the knowledge bottleneck.

We would also like to add more information about lexical semantics into the semantic representation so that we can add rules based on semantic classes. We are currently working on integrating English and Japanese WordNets for this purpose.

8 Conclusion

We presented an open-source Japanese–English machine translation system that contains both rule-based and statistical components. The rule-based translation engine uses a rich semantic representation (MRS) as a transfer language, allowing the development of powerful transfer rules that produce high-quality translations. By making the building blocks for semantic-transfer based MT available as open source, we aim to make substantive research in this framework possible for anyone.

Acknowledgements We would like to thank the members of the LOGON, Hinoki, and DELPHIN collaborations for their support and encouragement. In addition we would like to thank the developers and maintainers of the other resources we used in our project, especially JMDict, Tatoeba and Moses. This project was supported in part by the Norwegian Research Council (through support to the LOGON consortium and collaborators) and in part by Nanyang Technological University (through a start-up grant on "Automatically determining meaning by comparing a text to its translation").

References

Banerjee S, Lavie A (2005) METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, pp 65–72, URL `http://www.aclweb.org/anthology/W/W05/W05-0909`

Barreiro A, Scott B, Kasper W, Kiefer B (2011) OpenLogos machine translation: Philosophy, model, resources and customization. Machine Translation 25, (this volume)

Bond F, Breen J (2007) Semi-automatic refinement of the JMdict/EDICT Japanese-English dictionary. In: 13th Annual Meeting of the Association for Natural Language Processing, Kyoto, pp 364–367

Bond F, Oepen S, Siegel M, Copestake A, Flickinger D (2005) Open source machine translation with DELPH-IN. In: Open-Source Machine Translation: Workshop at MT Summit X, Phuket, pp 15–22

Bond F, Kuribayashi T, Hashimoto C (2008) Construction of a free Japanese treebank based on HPSG. In: 14th Annual Meeting of the Association for Natural Language Processing, Tokyo, pp 241–244, (in Japanese)

Bond F, Isahara H, Uchimoto K, Kuribayashi T, Kanzaki K (2010) Japanese WordNet 1.0. In: 16th Annual Meeting of the Association for Natural Language Processing, Tokyo, pp A5–3

Breen JW (2004) JMDict: a Japanese-multilingual dictionary. In: Coling 2004 Workshop on Multilingual Linguistic Resources, Geneva, pp 71–78

Burnard L (2000) The British National Corpus Users Reference Guide. Oxford University Computing Services

Callmeier U (2002) Preprocessing and encoding techniques in PET. In: Oepen S, Flickinger D, Tsujii J, Uszkoreit H (eds) Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing, CSLI Publications, Stanford, CA

Carroll J, Oepen S (2005) High-efficiency realization for a wide-coverage unification grammar. In: Dale R, Wong KF (eds) Proceedings of the 2nd International Joint Conference on Natural Language Processing, Lecture Notes in Artificial Intelligence, vol 3651, Springer, Jeju, Korea, pp 165 – 176

Copestake A (2002) Implementing Typed Feature Structure Grammars. CSLI Publications, Stanford, CA

Copestake A (2009) Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, pp 1–9

Copestake A, Flickinger D, Pollard C, Sag IA (2005) Minimal Recursion Semantics. An introduction. Journal of Research on Language and Computation 3(4):281 – 332

Dyvik H (1999) The universality of f-structure. Discovery or stipulation? The case of modals. In: Proceedings of the 4th International Lexical Functional Grammar Conference, Manchester, UK

Flickinger D (2000) On building a more efficient grammar by exploiting types. Natural Language Engineering 6 (1):15 – 28

Forcada ML, Ginestí-Rosell M, Nordfalk J, O'Regan J, Ortiz-Rojas S, Pérez-Ortiz JA, Sánchez-Martínez F, Ramírez-Sánchez G, Tyers FM (2011) Apertium: a free/open-source platform for rule-based machine translation. Machine Translation 25, (this volume)

Fujita S, Bond F, Oepen S, Tanaka T (2007) Exploiting semantic information for HPSG parse selection. In: Proceedings of the First ACL Workshop on Deep Linguistic Processing, Prague, Czech Republic, pp 25 – 32

Haugereid P, Bond F (2011) Extracting transfer rules for multiword expressions from parallel corpora. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World, ACL, Portland, Oregon, pp 92–100

Ikehara S, Shirai S, Bond F (1996) Approaches to disambiguation in **ALT-J/E**. In: International Seminar on Multimodal Interactive Disambiguation: MIDDIM-96, Grenoble, pp 107–117

Jellinghaus M (2007) Automatic acquisition of semantic transfer rules for machine translation. Master's thesis, Universität des Saarlandes

Koehn P, Shen W, Federico M, Bertoldi N, Callison-Burch C, Cowan B, Dyer C, Hoang H, Bojar O, Zens R, Constantin A, Herbst E, Moran C, Birch A (2007) Moses: Open source

toolkit for statistical machine translation. In: Proceedings of the ACL 2007 Interactive Presentation Sessions, Prague, URL http://www.statmt.org/moses/

Lardilleux A, Lepage Y (2009) Sampling-based multilingual alignment. In: Proceedings of Recent Advances in Natural Language Processing (RANLP 2009), Borovets, Bulgaria, pp 214–218

Mayor A, Alegria I, Díaz de Ilarraza A, Labaka G, Lersundi M, Sarasola K (2011) *Matxin*, an open-source rule-based machine translation system for basque. Machine Translation URL http://dx.doi.org/10.1007/s10590-011-9092-y

Mel'čuk I, Wanner L (2006) Syntactic mismatches in machine translation. Machine Translation 20:81–138

Nichols E, Bond F, Appling DS, Matsumoto Y (2007) Combining resources for open source machine translation. In: The 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07), Skövde, pp 134–142

Nichols E, Bond F, Appling DS, Matsumoto Y (2010) Paraphrasing training data for statistical machine translation. Journal of Natural Language Processing 17(3):101–122, special Issue on Empirical Methods for Asian Language Processing

Nygaard L, Lønning JT, Nordgård T, Oepen S (2006) Using a bi-lingual dictionary in lexical transfer. In: Proceedings of the 11th Conference of the European Association for Machine Translation, Oslo, Norway

Och FJ (2005) Statistical machine translation: Foundations and recent advances. In: MT Summit X Tutorial, Phuket

Och FJ, Ney H (2002) Discriminative training and Maximum Entropy models for statistical machine translation. In: Proceedings of the 40th Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp 295 – 302

Oepen S, Flickinger DP (1998) Towards systematic grammar profiling. Test suite technology ten years after. Journal of Computer Speech and Language 12 (4) (Special Issue on Evaluation):411 – 436

Oepen S, Lønning JT (2006) Discriminant-based MRS banking. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Genoa, Italy

Oepen S, Dyvik H, Lønning JT, Velldal E, Beermann D, Carroll J, Flickinger D, Hellan L, Johannessen JB, Meurer P, Nordgård T, Rosén V (2004a) Som å kapp-ete med trollet? Towards MRS-based Norwegian – English Machine Translation. In: Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation, Baltimore, MD

Oepen S, Flickinger D, Toutanova K, Manning CD (2004b) LinGO Redwoods. A rich and dynamic treebank for HPSG. Journal of Research on Language and Computation 2(4):575 – 596

Oepen S, Velldal E, Lønning JT, Meurer P, Rosén V, Flickinger D (2007) Towards hybrid quality-oriented machine translation. On linguistics and probabilities in MT. In: Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation, Skövde, Sweden

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: 40th Annual Meeting of the Association for Computational Linguistics: ACL-2002, pp 311–318

Paul M (2006) Overview of the IWSLT 2006 Evaluation Campaign. In: Proc. of the International Workshop on Spoken Language Translation, Kyoto, Japan, pp 1–15

Siegel M, Bender EM (2002) Efficient deep processing of Japanese. In: Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics, Taipei, pp 1–8

Sukehiro T, Kitamura M, Murata T (2001) Collaborative translation environment 'Yakushite.Net'. In: Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium: NLPRS-2001, Tokyo, pp 769–770

Tanaka Y (2001) Compilation of a multilingual parallel corpus. In: Proceedings of PACLING 2001, Kyushu, pp 265–268, (http://www.colips.org/afnlp/archives/pacling2001/pdf/tanaka.pdf)

Uchimoto K, Zhang Y, Sudo K, Murata M, Sekine S, Isahara H (2004) Multilingual aligned parallel treebank corpus reflecting contextual information and its applications. In: Sérasset G (ed) COLING 2004 Multilingual Linguistic Resources, COLING, Geneva, Switzerland, pp 57–64, URL http://acl.ldc.upenn.edu/W/W04/W04-2208.bib

Velldal E, Oepen S (2006) Statistical ranking in tactical generation. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia

Way A (1999) A hybrid architecture for robust MT using LFG-DOP. Journal of Experimental and Theoretical Artificial Intelligence 11:441–471, special Issue on Memory-Based Language Processing