Constraint Grammar-Based Swedish-Danish Machine Translation

Eckhard Bick

Institute of Language and Communication, University of Southern Denmark,
Campusvej 55, DK-5230 Odense M, Denmark
eckhard.bick@mail.dk

Abstract. This paper describes and evaluates a grammar-based machine translation system for the Swedish-Danish language pair. Source-language structural analysis, polysemy resolution, syntactic movement rules and target-language agreement are based on Constraint Grammar morphosyntactic tags and dependency trees. Lexical transfer rules exploit dependency links to access contextual information, such as syntactic argument function, semantic type and quantifiers, or to integrate verbal features, e.g. diathesis and auxiliaries. Out-of-vocabulary words are handled by derivational and compound analysis with a combined coverage of 99.3%, as well as systematic morpho-phonemic transliterations for the remaining cases. The system achieved BLEU scores of 0.65-0.8 depending on references and outperformed both STMT and RBMT competitors by a large margin.

Keywords: Machine Translation, RBMT, Constraint Grammar.

1 Introduction

Over the last decade, riding on an exponential growth curve of computer processing power and corpus size, Statistical Machine Translation (STMT) has outpaced research into Rule-Based Machine Translation (RBMT), albeit there is a certain interest in hybrid systems, not least for languages with a rich morphology and a need for syntactic reordering (e.g. Hindi, Ahsan et al. 2010). Since STMT is a machine learning technique that depends on the availability of (bilingual or comparable) training data, it has enormously profited from big data techniques in general, but while regular parallel corpora like Europarl (Koehn 2005) have helped to develop the necessary methods, the largest public success has been achieved by the lords of big data, Internet giants Google and Bing, which can now be seen as bench marks for less widely known, and more specialized systems. Given the necessary data trove, STMT is a very cost-efficient method to produce machine translation for many language pairs and to harvest fluency from target language examples. However, STMT still suffers from certain more or less inherent problems:

1. For less-resourced languages, STMT may lack sufficient training data. This is particularly true if both languages in a translation pair are small. Even with English-bilingual data at hand, there will be quality loss in an English-mediated

A. Przepiórkowski and M. Ogrodniczuk (Eds.): PolTAL 2014, LNAI 8686, pp. 216–227, 2014.

transfer, because English as an interlingua[1] may disguise or create ambiguities relevant to the languages in question.

2. Without access to systematic linguistic analysis and generation, STMT on its own has difficulties in handling morphologically rich languages (Ahsan et al. 2010), because the individual inflexions, derivations and compounds are too rare. In addition, without syntactic-structural analysis long-distance features such as agreement between clause parts, pronominal anaphora and reflexivity are out of reach for n-gram based machine learning.
3. Because it is not possible to interfere directly with a phrase-based STMT core module, it is difficult to fix individual or systematic errors, even when identified, or to pass lexical knowledge such as word sense disambiguation on to an STMT system (Carpuat & Wu 2005), and it is not possible to systematically adapt the system to a domain different from the ones training data is available for.
4. Without dictionaries in a more traditional sense, STMT systems run the risk of semantic confusion of words that share the same context, such as currency words and antonyms.

Rule-based systems, on the other hand, can implement symbolic language models for small languages even in the absence of large corpus data (1), both for analysis and transfer, as argued for instance by Seiss & Nordlinger (2001) who use morphological finite-state transducers and manual transfer rules for Murrinh-Patha. Rule-based MT systems (as well as certain hybrid combinations) support deep module integration and have system-wide access to a full linguistic analysis, allowing them to take into account both analytical morphology and long-distance relations (2). Needless to say, rules can be changed or amended “locally” in order to handle errors, sense distinctions or domain migration (3). And finally, in a manual translation lexicon, currency words and antonyms will be listed individually without the risk of contextual confusion (4).

We therefore believe that rule-based MT systems should be given a second chance, not least for less-resourced languages, and though hybrid systems may be the ultimate solution, relatively “pure” systems can demonstrate strengths, weaknesses and evaluative insights that may guide later hybridization efforts. We are aware that phrase-based STMT can be improved by adding higher-order linguistic relations, for instance manual reordering rules using dependency relations for language pairs with SVO/SOV word order differences (Peng Xu et al. 2009), but it would be an added advantage if such pre- and postprocessing modules could be handled in one formalism, with a shared category set, shared tags and shared lexical information, with the possibility of easy cross-reference between modules, and an integrated system architecture. In this paper we present such a system for two small languages (Swedish to Danish), implementing a rule-based approach relying on high-quality

[1] English is more distant from both Swedish and Danish than the two languages are from each other. Rather, if the training data problem can be overcome through RBMT, it would make more sense to use one Scandinavian language as interlingua for another, as proposed by Bick & Nygaard (2007) for Norwegian and Danish.

source language dependency analysis (Constraint Grammar, Bick 2005) as a matrix for anchoring context-driven transfer rules, disambiguation, agreement and compounding. The general architecture of this approach is in principle language-independent, and inspired by similar work for the Danish-English machine translation (Bick 2007).

2 System Architecture

On the source language side, the core of the system is SweGram,[2] a Constraint Grammar (CG) parser using the CG3[3] formalism and rule compiler. SweGram was designed with robust corpus annotation[4] in mind, and provides the following information in a tag-based fashion, based on a 70,000-lemma lexicon and 8,500 tagging and disambiguation rules:

1. tokenization, including abbreviations, numerical and scientific expressions, complex function words and named-entity recognition (NER)
2. morphological analysis, including compound recognition, derivation and endings-based out-of-vocabulary heuristics
3. syntactic function tags (subject, object, predicative etc.)
4. dependency trees
5. semantic classification of common and proper nouns and valency classification of verbs

For instance, in the example analysis below, word #13, “snösmältningsmaskin” (snow-melting machine) is recognized as an allowed compound (with a fuge-s), and tagged as a noun (N) singular (S) in the common gender (UTR), nominative (NOM) and indefinite (IDF), functioning as the head of a direct/accusative object (@ACC) of word #10, the verb “inviga” (introduce, take into use).

De	[den] <*> ART nG P DEF @>N #1->3
senaste	[sen] <jtemp> ADJ SUP nG nN DEF NOM @>N #2->3
dagarnas	[dag] <dur> <temp> N UTR P DEF GEN @>N #3->4
snöstorm	[snöstorm] <event> <wea> <F:snö+storm> N UTR S IDF NOM @SUBJ> #4 >7
i	[i] <np-close> PRP @N< #5->4
New=York	[New=York] <civ> <*> PROP NEU S NOM @P< #6->5
fick	[få] <vt+INF> <mv> V IMPF AKT @FS-STA #7->0

[2] A demo version of the parser, as well as an overview of category and tag definitions, can be accessed at `http://beta.visl.sdu.dk/visl/sv/parsing/automatic/`

[3] CG3 has been developed as an open-source tool by the Danish language technology company GrammarSoft ApS in cooperation with the University of Southern Denmark, who maintain a documentation and download site at `http://beta.visl.sdu.dk/constraint_grammar.html`.

[4] SweGram annotated corpora are accessible at `http://corp.hum.sdu.dk` (CorpusEye)

myndigheterna	[myndighet] <HH> <aci-subj> N UTR P DEF NOM @<ACC #8->10
att	[att] INFM @INFM #9->10
inviga	[inviga] <mv> V INF AKT @ICL-<OA #10->7
en	[en] ART UTR S IDF @>N #11->13
splitterny	[splitterny] <heur> <F:splitter+ny> ADJ UTR S IDF NOM @>N #12->13
snösmältningsmaskin	[snösmältningsmaskin] <good-compound <N:snösmältning~s+maskin> <mach> N UTR S IDF NOM @<ACC #13->10
$.	[.] PU @PU #14->0

[1-The 2-last 3-days' 4-snow storm 5-in 6-New York 7-got 8-the authorities 9-to 10-take into use 11-a 12-brand new 13-snow melter]

Note that this 13-word sentence contains 3 compounds, which is quite normal for Swedish. Though only one (“snöstorm”) was listed in the parser's lexicon, it assigned correct lexical types (<wea> <event> weather event, and <mach> machine) to the nouns. This will allow the translation system to construct plausible translations for the compounds, and provide useful semantic context to other words in the sentence. The parser also identified the special construction “få till att” (get sb to do sth), marking “myndigheterna” (authorities) as both object of “få” and subject (<aci-subj>) of “inviga”, and allowing the translator to pick a reasonable translation for “få” (get) which in Swedish is just as phrasally ambiguous as in English.

3 Lexical Transfer

For lexical transfer, Swe2dan exploits CG tags in two ways, as either local or contextual distinctors, which are used to formulate transfer rules that help the system decide which translation to pick, and can be used both for polysemy resolution and usage differences (“synonym picking”), with no statistical element needed. The idea is not to define, but to distinguish meanings. While local distinctors refer to tags on the token itself (e.g. part of speech, number, syntactic function, domain), contextual distinctors refer to features of arguments, attributes, heads etc. of the word in question, tracing dependency links to the second degree, or simply using relative positions left and right. The function word “när” *(near, when)*, for instance, has 5–6 different translations, not counting fixed expressions, which go in a separate lexicon. However, the different translations can be reliably distinguished by word class (adverb ADV vs. conjunction KS), clause type (interrogative vs. relative), syntactic function (pre-adject @>A), head verb tense (IMPF) or immediate left context (P-1)[5].

- när_ADV :nær *[near]*; S=(<interr>) :hvornår *[when?]*; S=(<rel>) :når *[(at the time) when]*; S=(@>A) :næsten *[almost]*; P-1=('hart') :næsten *[almost]*
- när_KS :når *[when(ever)]*; H=(IMPF) :da *[when ..ed]*

[5] A reference to the head verb is clearly a long-distance distinctor, but even local distinctors may depend on larger contexts – thus, syntactic function and clause type are local only in the sense that the CG engine has created local tags for them.

In spite of the relatedness of Swedish and Danish, a one-on-one translation is possible in less than 50% of all tokens. Thus, lexicon entries with transfer rules account for only 4% of the ca. 100,000 lexemes, but for 53% in frequency terms. In other words, frequent lexemes are much more ambiguous, and more prone to usage variation than rarer ones. The structurally most important word class, verbs, stands for 40% of all contextual transfer rules. In the example, 5 translations for the verb “fräsa” are distinguished by specifying daughter-dependents (D) or dependents of dependents (granddaughters, GD) as subjects (@SUBJ) or objects (@ACC) with certain semantic features, such as human <H>, vehicles <V> or <food>. For closed-class items such as prepositions or adverbs (here: “åt”, “iväg”, “förbi”), it often makes sense to refer directly to word forms. Negative conditions are marked with a '!'-sign, optional conditions with a '?'.[6]

fräsa_V :hvæse *(to hiss like a cat);*
D1=("åt") GD1=(<H>) D2=(<H> @SUBJ) :vrisse *(to snap at sb)*;
D=(<[HV].*> @SUBJ) D=("(iväg|förbi)") D!=(@ACC) :rase *(tear/speed along)*;
D=(<food.*> @ACC) :stege, :brune, :brase, :lynstege *(to fry)*;
D=(@ACC) D=(<H> @SUBJ) :fræse *(to mill, to cut a material or tool);*

Sometimes a distinction depends on clues that are present in the overall context, but have not been explicitly tagged by the SweGram parser. We therefore introduced a separate CG, run after the parser and before translation, that adds the desired tags from context. Examples are reflexivity, article insertion or the propagation of number, definiteness and the +human feature to under-specified heads or dependents, or from anaphoric referents to pronouns. A second round of feature propagation is done in the translation program itself, after translations have been chosen. For instance, if a translated Danish noun differs from the Swedish original in gender (or sometimes, number), all other members of the noun phrase need to have changed their gender, too.

Finally, compounding and affixation can be used to assign different translations depending on whether a lexical item is used as first, last (second) or middle part, if necessary in combination with further conditions:

lock_N (25) :lok, :hårlok *[curl]*; S=(<second>) :låg *[cover]*; S=(NEU) :låg; S=(< first>) :lokke *[luring]*

4 Out-of-Vocabulary Words

Out-of-vocabulary words can occur both at parser level and translation level. Though we are primarily concerned with the latter here, it is relevant that the SweGram parser normalizes simple misspellings and provides a compound analysis for both known

[6] The formalism also allows reference to heads (H), heads of heads (grandmothers, GM), to numbered relative positions and their dependents, dependency direction, dependent n-grams and others.

and unknown compounds. As a first fallback for out-of-vocabulary words, the translator performs part-for-part translations following the parser's compound break-down, using the above-mentioned rules for first and second parts. The second fallback is *transformation* rather than translation, i.e. heuristic translation based on partial translation (of word endings and affixes) and systematic letter replacement. For instance the definite plural noun ending '-orna' will be changed into Danish '-erne', the affixes '-ism' and '-skap' become '-isme' and '-skab', and Swedish 'ö'/'ä' will become Danish 'ø'/'æ'. Similar changes apply for participle and verb endings, as well as consonant gemination. In a way, the rationale behind these changes is treating Swedish as a kind of misspelled Danish, exploiting that a large portion of words has a common etymology.

In a newspaper test corpus with 144,456 non-punctuation tokens the parser classified 7,120 unknown non-name words as "good compounds" and 1,245 as outright heuristic analyses. Swe2dan came up with non-heuristic translations for 99.1% of the compounds and had ordinary lexicon entries for 62.1% of the heuristicals, leaving the rest to the transformation module. For ordinary, parser-sanctioned words the translation lexicon had a coverage of 99.71%, missing out on only 368 words, and bringing total coverage up to an impressive 99.33%. A break-down of the 368 words that were known to the parser, but not to the translator, showed that roughly half (51.6%) of the Swedish words, many of them foreign, were left as-is and worked in Danish, too. 11.7 percent were transformed into the correct Danish word, and 36.7% produced wrong translations, either unchanged or with wrong or partial transformations. Very few good-compounds needed transformational translation, and in only 2 cases the transformation was wrong. 69% of the out-of-vocabulary words deemed "heuristic" by the parser were misspellings, non-letter characters and word fusions due to missing spaces. 11% had transformations (10% wrong, 1% correct), and 10% were correctly left as is. Including correct transformations and as-is translations, overall translation coverage was 99.62%.

5 Target Language Generation

Swedish and Danish both have 2 genders, but depending on the translation, noun genders may not match, not even for cognates. While finding the correct inflection for the noun itself is a simple lookup procedure, it is more difficult to propagate gender and number changes to the whole np, predicatives and complements which may be located far away in the sentence. Therefore, even the generator profits from the deep CG parse, propagating inflection changes along dependency links.

Evaluated on a 10,000 sentence newspaper text chunk, the coverage of the Danish generation lexicon was satisfactory. 94.49% of lookups were successful as complete lexemes, for 3.03% compound analyses provided by the parser were used for lookup of the inflectionally relevant last part of the compound, and for 0.78% the generator itself was able to create a (heuristic) compound analysis. As a last resort, unknown words were inflected following the most common Danish paradigm for the word class in question. This was necessary in 1.84% of cases (including 0.14% related to compound analysis), but caused virtually no inflection errors in the examined sample, probably because irregular forms tend to be frequent (and therefore lexicon-covered), while it is the regular paradigms that are productive and cover most of the Zipf curve tail. With a combined coverage of almost 100%, errors from the TL generation module are unlikely to be lexically caused, leaving only errors caused by wrongly assigned inflection tags, which are difficult to isolate from SL analysis and transfer.

6 Structural Transfer

Swedish and Danish are both East-Scandinavian languages and share basic sentence structure. However, there are certain important differences the treatment of which asks for a scope beyond n-gram matches:

1. Due to certain differences in the expression of definiteness, an np's left (article or pronoun) and right (noun inflection) edges have to be handled interdependently.
2. Adverb position differs on several accounts: Adverb particles of transitive phrasal verbs are placed after the object in Danish, but before it in Swedish.
3. Swedish has special supine verb forms, many of which are morphologically indistinguishable from active voice participles, which have to be translated with either past tense verbs or participle constructions in Danish.

At the level of individual words, our MT system can handle these cases in the lexical transfer rules themselves. For instance, the translation of the definite article (1) may be set to “nil” in a context of H=(N DEF), i.e. a definite head noun (Swedish double definiteness). Where more elaborate, or more global, conditions are needed, special CG rules are used. Like the underlying parser, these CG rules have access to virtually all tags and relations, and can change definiteness inflection, or insert articles, in preparation of the Danish translation. (2) can in principle also be handled by transfer rules, setting the translation of a phrasal adverb as “nil"” while at the same time adding it to the object translation (a kind of indirect movement instruction):

packa_V D=("ut")_nil D=(@ACC)_[+ud] :pakke *[unpack – "pack out"]*

However, the lexical load is much bigger in this case. We therefore (also) use dependency-based movement rules like the following:

w(@MV<|@OA),g(<right> @ACC) -> 2,1

This rule changes the order of a word constituent (w) that is a verb particle (@MV<) or adverbial object complement (@OA) with a group constituent (g) that is a direct object (@ACC) to the right of its head verb. The rule will work independently of the size of the object np, because all dependents are automatically included in the movement. Similar adverb movement rules are also necessary for inverted (VS) clause order, where adverbs have to be moved out of the VS bracket (V A S → V S A), or for infinitive markers (insertion or adverb movement). All in all, the movement grammar contains 61 rules.

Supine forms, finally, (3) need structural information twice – first, the parser needs global context to disambiguate the form itself (it is ambiguous with ordinary, np-internal participles for all regular verbs), second, translation tense has to be chosen, with the possible insertion of an auxiliary and corresponding adverb or subject movements.

7 Evaluation

We evaluated the system on 100 random new sentences (ca. 1,500 words), taken from the Leipzig Wortschatz corpus collection,[7] comparing GramTrans' Sve2dan translations to those of three other systems, Google Translate,[8] Bing Translator[9] and Apertium,[10] all of which maintain open-access user interfaces. While Google Translate and Bing Translator rely on STMT, Apertium (Tyers et al. 2010) is an open-source RBMT system like GramTrans itself. However, where Apertium uses corpus-trained HMM taggers, GramTrans is rule-based also in the SL analysis modules.

First, we measured all systems against both an independent manual translation and best-case edited system translations, using the BLEU (Papineni et al. 2002) and NIST metrics.

Table 1. BLEU/NIST scores

	Manual reference (1)	**Edited system reference**	**Multi-reference (all minus self)**
GramTrans	**0.645** / 8.515	**0.838** / 9.817	**0.757** / 10.050
Google	0.387 / 6.300	0.645 / 8.361	0.539 / 8.150
Apertium	0.390 / 6.391	0.516 / 7.361	0.468 / 7.418
Bing	0.342 / 6.006	0.600 / 8.064	0.492 / 7.793

In this comparison, GramTrans clearly outperformed all other systems. Apertium performed slightly better than the statistical systems, when measured against one manual translation, but came out last when measured against "self-edit" or "all others".[11] The statistical systems profited relatively more from the inclusion of self-edits, and in relative

7 `http://corpora.informatik.uni-leipzig.de/download.html`
8 Translations were taken from `http://translate.google.com/` (15 March 2014)
9 Accessed at `http://www.bing.com/translator` (15 March 2014)
10 We used the demo at: `http://www.apertium.org/?lang=n&lang=en` (15 March 2014)
11 In their own evaluation of Swedish-Danish Apertium and an early unpublished version of GramTrans, Tyers & Norfalk (2009), using word edit rates (WER) and editing distance, ranked their system (WER 30) below GramTrans (WER 26) and Google (WER 35), even when introducing an anti-GramTrans bias by only measuring GramTrans against post-edited Apertium-translations.

terms the difference between GramTrans and Google was bigger for BLEU than for NIST in all runs. Since NIST downplays the importance of short/common words and of small length difference, this finding may result from a particular strength of rule-base systems – function words, definiteness and inflexion/agreement, all of which result in the kind of "short" differences under-weighted by NIST.

In order to determine system similarity, we also measured systems against each other, using one system's edited translation as a BLEU-reference for another system:

Table 2. Cross-system BLEU scores

Reference: Test:	GramTrans edited	Google edited	Apertium edited	Bing edited
GramTrans	(0.838)	0.497	**0.666**	0.501
Google	0.387	(0.645)	0.384	**0.478**
Apertium	**0.426**	0.330	(0.516)	0.325
Bing	0.358	**0.446**	0.353	(0.600)

Here, GramTrans and Apertium score slightly better against each other's edited versions than against the manual standard, while dropping against Google and Bing edits, attesting to similar (rule-based) translation styles. Likewise, the statistical systems Google and Bing perform better against each other than against either the manual translation or edits of the rule-based systems.

Expecting even clearer correlations between results and RBMT/SMT system styles, we also performed an edit distance evaluation, using the TER metric (*Translation Error Rate,* Snover 2006) and again comparing system translations with both manual and edited translations. In TER, error rates can be interpreted as editing distances, covering insertion, deletion or substitution of a word, or the movement (shift) of a word or word chain. Fewer edits mean a lower TER and a better performance.

Table 3. TER distances

Reference: Test:	Manual	GramTrans edited	Google edited	Apertium edited	Bing edited
GramTrans	**20.84**	**(8.57)**	32.12	***19.77***	30.98
Google	45.05	44.40	(23.60)	45.20	***37.56***
Apertium	34.54	***31.13***	41.96	(24.51)	41.75
Bing	48.62	46.98	***40.70***	48.03	(28.05)

Again, GramTrans outperformed its competitors. Its relative advantage compared with the statistical systems was even bigger with TER than with BLEU, and it had the lowest editing cost (scores in parentheses), i.e. the GramTrans-translation needed fewest changes to become lexico-grammatically acceptable. In addition, GramTrans was also, for all combinations, the cheapest system to turn into the edited version of another system. Against the manual translation, GramTrans' relative TER advantage is shared by the other RBMT system, Apertium, which in the TER evaluation ranked second against this reference, while performing similar to Google with BLEU. Cross-system editing distances indicate that the statistical pair on the one hand (Google and Bing), and the rule-based pair on the other (GramTrans and Apertium) share inherent features even in the edited versions (bold italics).

A break-down of edit types corroborates the difference between the two system types:

Table 4. TER evaluation, edit types

	Insertions	**Deletions**	**Substitutions**	**Shifts (word shifts)**	**Ins&del / subs**	**TER**
GramTrans	20	11	103	5 (6)	0.40	8.57
Google	71	51	263	11 (11)	0.46	23.60
Apertium	21	33	333	11 (13)	0.16	24.51
Bing	87	74	290	18 (19)	0.56	28.05

The STMT systems have a relatively high need for deletions and insertions, compared to substitutions (second-last column), a finding that might be linked to problems with small function words and articles, possibly compound splitting caused by using English translations of Swedish compounds as a fall-back route into Danish (which compounds the same way Swedish does). Apertium has the opposite problem, with a high proportion of substitutions (in part due to its low lexical coverage), but a good score for insertions/deletions. Bing sticks out with the highest need for movements, indicating a poor syntactic engine.

A qualitative inspection of the data showed that GramTrans (and Apertium) completely avoided a number of error types typical of STMT systems:

- the confusion of “ontological sister terms”
  - Bing: “dollar/kroner”
  - Bing: “Per Wesslén/Wade”, “Hale/Halestone”, “Svensson/Smith”
  - Google: “Solbergaskolen”/”Solbergaleden”
- the literal translation of names caused by case folding
  - Bing: “Huge chockstartade” → “Kæmpe [= big/huge] chockstartade”

- errors caused by using a big-data training language as an intermediate step between small languages with insufficient direct training data,
  - Google: “styrelse” (Swedish) → “board” (English) → “bord” [table] instead of “bestyrelse” (Danish)
  - compounding: Google: “säkerhetsexpert” (Swedish) → “security expert” (English) → “sikkerhed ekspert” instead of “sikkerhedsekspert” (Danish)
  - Google: “Se upp för elgen” (Swedish) → “Watch out for elg” (Danish)

Another difference were long-distance syntactic relations, such as the choice between reflexive (“sin”) and non-reflexive (“hans”) possessive pronouns, which were handled well by the RBMT systems, but badly by the STMT systems on all occasions. In the example sentence the reflexive noun phrase is correctly translated by GramTrans, while Bing introduces errors in both reflexivity (refl), number and compounding[12]. Google gets the number feature right, but not the other two, while Apertium simply retains the Swedish expression, including the reflexive.

Table 5. Qualitative differences: Reflexives

		refl	**number**	**lex**	**compound**
Swedish original	Nicole talar ut om sina viktproblem *Nicole speaks out about her weight problems*				
GramTrans	Nicole taler ud om sine vægtproblemer	ok	ok	ok	ok
Google	Nicole taler ud om hendes vægt problemer	err	ok	ok	err
Apertium	Nicole taler ud om sine viktproblem	ok	(ok)	err	(ok)
Bing	Nicole taler ud om hans vægt problem	err	err	ok	err

8 Conclusion and Outlook

We have shown, for the Swedish → Danish language pair, that a modular CG-based translation system with manual transfer rules can outperform bench mark systems such as Google Translate and Bing Translator, and discussed the architecture and performance of the individual modules. The system is publicly available at http://gramtrans.com, and being used in an integrated Swedish-Danish version of Wikipedia (http://dan.wikitrans.net). Future research goals include improved heuristics for out-of-vocabulary words, domain flags and improvements to a fledgling Danish → Swedish sister system – a task that for RBMT is by no means trivial since neither SL analysis nor transfer and disambiguation rules can be reused.

[12] Using the ordinary possessive forces systems to decide on possessor gender, and unlike Google, Bing gets that wrong, too.

References

1. Ahsan, A., Kolachina, P., Kolachina, S., Sharma, D.M.: Coupling Statistical Machine Translation with Rule-based Transfer and Generation. In: Proceedings of the 9th Conference of the Association for Machine Translation in the Americas, Denver, Colorado (2010), http://amta2010.amtaweb.org/AMTA/papers/2-06-AhsanKolachinaEtal.pdf
2. Bick, E.: Dan2eng: Wide-Coverage Danish-English Machine Translation. In: Maegaard, B. (ed.) Proceedings of Machine Translation Summit XI, Copenhagen, Denmark, September 10-14, pp. 37–43 (2007)
3. Bick, E., Nygaard, L.: Using Danish as a CG Interlingua: A Wide-Coverage Norwegian-English Machine Translation System. In: Nivre, J., Kaalep, H.-J., Muischnek, K., Koit, M. (eds.) Proceedings of the 16th Nordic Conference of Computational Linguistics, May 24-26, pp. 21–28. University of Tartu (2007)
4. Bick, E.: Turning Constraint Grammar Data into Running Dependency Treebanks. In: Civit, M., Kübler, S., Martí, M.A. (eds.) Proceedings of TLT 2005, pp. 19–27. Universitat de Barcelona (2005)
5. Carpuat, M., Wu, D.: Word Sense Disambiguation vs. Statistical Machine Translation. In: Knight, K., Ng, H.T., Oflazer, K. (eds.) Proceedings of ACL 2005, pp. 387–394. Association for Computational Linguistics, Stroudsburg (2005)
6. Koehn, P.: Europarl: A Parallel Corpus for Machine Translation. Machine Translation Summit, pp. 79–86 (2005)
7. Papineni, K., et al.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th ACL, Philadelphia, pp. 311–318 (July 2002)
8. Seiss, M., Nordlinger, R.: An Electronic Dictionary and Translations System for Murrinh-Patha. In: Gimeno, A. (ed.) Proceedings of EUROCALL 2011, vol. 20(1) (2012); The Eurocall Review
9. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223–231 (2006)
10. Tyers, F.M., Sánchez-Martínez, F., Ortiz-Rojas, S., Forcada, M.L.: Free/open-source resources in the Apertium platform for machine translation research and development. The Prague Bulletin of Mathematical Linguistics (93), 67–76 (2010)
11. Tyers, F.M., Nordfalk, J.: Shallow-transfer rule-based machine translation for Swedish to Danish. In: Pérez-Ortiz, J.A., Sánchez-Martínez, F., Tyers, F.M. (eds.) Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, pp. 27–33. Universidad de Alicante, Alicante (2009)
12. Xu, P., Kang, J., Ringgaard, M., Och, F.: Using a dependency parser to improve SMT for subject-object-verb languages. In: Proceedings of NAACL 2009, pp. 245–253. Association for Computational Linguistics, Morristown (2009)