Rule-based Machine Translation from English to Finnish

Arvi Hurskainen
University of Helsinki
arvi.hurskainen@helsinki.fi

Jörg Tiedemann
University of Helsinki
jörg.tiedemann@helsinki.fi

Abstract

The paper describes a rule-based machine translation system adapted to English to Finnish translation. Although the translation system participates in the shared task of news translation in WMT 2017, the paper describes the strengths and weaknesses of the approach in general.

1 Credits

We are grateful to Pasi Tapanainen from Connexor OY for allowing us to use the en-fdg analyser of English as well as the Constraint Grammar (CG-3)[1] environment for a number of translation phases.

2 Introduction

The translation system described here is in stark contrast to the majority of systems participating in this conference. There are a number of reasons why we are interested in developing rule-based translation systems. One is the observation that, if we use statistical or neural translation systems, we will exclude 99.8 percent of languages out of development. Digitalization is supposed to break barriers between language groups, but in fact it currently increases them. The current hype on neural methods still accelerates the break between the small group of dominant languages and the less-resourced ones. If we want to avoid the break, we do not see any other way out than to put efforts in developing such systems that are affordable for less-resourced languages. At the same time, efforts for finding ways to overcome the problems of statistical systems are needed (Tiedemann et al 2016).

The approach described here deals with the English to Finnish translation system. However, the basic components of the system were developed with the language pair Swahili and English, for which Hurskainen developed rule-based translation systems to both directions (Hurskainen 1992, 1996, 2004, 2006, 2007, 2012).

A number of approaches for constructing rule-based systems have been studied. These include OpenLogos (Scott and Barreiro 2009; Barreiro et al 2011), Apertium (Forcada 2006), Grammatical Framework (Ranta 2011), and Nooj (Silbertztein 2015). Common to these approaches is the use of grammatical knowledge and lexicon of languages in translation. Although the approach that we have used has much in common with those, we did not implement any of them directly. The main reason is that we find it useful to have full control of all phases of the translation process, so that corrections can be made instantly at the correct point of the process. For the same reason we did not adapt such resources as Omorfi (Pirinen 2015). Instead we developed our own system for generating Finnish word forms.

The system described here deliberately avoids any statistical elements in translation process. The basic assumption is that running text can always be decomposed into structured units, and that these units can be described on more or less general level. The translation is not performed on the basis of surface word forms, but rather as a controlled sequence of operations, where the text in source language is processed into surface form of the target language. The basic components in the system are the lexicon and grammar of both languages.

On the abstract level, the language can be described by means of tags, each of which represents various degrees of abstractness. For example, POS tags are the most abstract ones, each representing a large set of members, whereas word lemmas are least abstract, and morphological tags are somewhere in between. The combination of the tags constitutes the knowledge, on the basis of which the text is converted into the surface form of the target language.

There are two guiding principles in this translation system. First, each word form should be given all linguistically correct interpretations. Second, all such operations that are conditional of context, such as selection, deletion, replacement, and adding, should be done in the environment, where context-sensitive rules can be written for controlling the process. For this rea-

[1] CG-3 is also termed as FDG-3, because within this environment it is possible to write also functional dependency rules.

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 323–329
Copenhagen, Denmark, September 711, 2017.

son, Constraint Grammar (Karlsson 1990, 1995; Karlsson et al 1995; Tapanainen 1996; Bick and Didriksen 2015) is in important role in the system.

Below is a description of various phases of the translation process.

3 Analysis of source text

The source text is first morphologically analysed, disambiguated and provided with syntactic description. In analysing the English text, we used the en-fdg parser (Järvinen and Tapanainen 1997). The parser has a fairly covering vocabulary, and it performs surface-syntax parsing as well as dependency parsing. However, it makes mistakes, and wrong assignments especially in POS categories are detrimental to translation results. Since we had no access to the source code of the parser, we had to devise our own mechanism to correct the mistakes.

The example sentence in (1) is used throughout in this paper.

(1)
```
1 He he subj:>2 @SUBJ %NH PRON PERS NOM SG3
2 will will v-ch:>3 @+FAUXV %AUX V AUXMOD
3 be be v-ch:>4 @-FAUXV %AUX V INF
4 hanging hang main:>0 @-FMAINV %VA ING
5 out out phr:>4 @ADVL %EH ADV
6 on on loc:>4 @ADVL %EH PREP
7 stages stage pcomp:>6 @<P %NH N NOM PL
8 for for subj:>11 @ADVL %EH PREP
9 years year pcomp:>8 @<P %NH N NOM PL
10 to to pm:>11 @INFMARK> %AUX INFMARK>
11 come come mod:>7 @-FMAINV %VA V INF
```

The en-fdg parser performs two types of syntactic mapping, and we had to choose one of them. Because the rule system, which we were going to use in the translation system, makes use of relative distances, we decided to use the surface-syntax option. The precise distances that the dependency parsing produces would probably not have much helped in translation. The modified form is in (2).

(2)
```
"<*he>" "he" %SUBJ CAPINIT PRON PERS NOM SG3
"<will>" "will" %+FAUXV V AUXMOD
"<be>" "be" %-FAUXV V INF
"<hanging>" "hang" %-FMAINV ING
"<out>" "out" %ADVL ADV
"<on>" "on" %ADVL PREP
"<stages>" "stage" %<P N PL NOM
"<for>" "for" %ADVL PREP
"<years>" "year" %<P N PL NOM
"<to>" "to" %INFMARK> INFMARK>
"<come>" "come" %-FMAINV V INF
```

4 Isolation of multiword expressions

Multiword expressions (MWE) are becoming an increasingly important component in machine translation. There is no covering list of MWEs of English, because the concept is very fluid. Many clusters of words can be successfully treated in more than one way. The general rule is that if translation through the normal rule system does not succeed, consider treating the cluster as a MWE. Treating a structure, which also could be handled with normal rules, as a MWE, often helps in disambiguation, because the MWE is given the lexical representation in target language for all members of the structure. In general, it is more safe to use MWE treatment is cases where both options are possible,

For this reason, MWEs are isolated prior to inserting the glosses (i.e. lexical words) of the target language. These MWEs are given the appropriate lexical interpretation (3).

(3)
```
"<*he>" "he" %SUBJ CAPINIT PRON PERS NOM SG3
"<will>" "will" %+FAUXV V AUXMOD
"<be>" "be" %-FAUXV V INF
"<hanging_out>" "hang_out" { hengailla V67 ,
roikkua V52-A } %-FMAINV ING
"<on>" "on" %ADVL PREP
"<stages>" "stage" %<P N PL NOM
"<for_years_to_come>" "for_year_to_come" {
tulevina vuosina , tuleviksi vuosiksi } ADV
```

We have used the CG-3 rule formalism (Tapanainen 1996) for implementing the MWEs, because it has a sophisticated system for controlling the rule application on the basis of context. It also removes all grammatical information on words, which are not relevant in further processing.

In English to Finnish MT, it is sensible to identify four types of MWEs, (a) those which have no inflection, (b) those where the first element (noun) carries the information on inflection, (c) those where the last element (noun) carries the information on inflection, and (d) those where the verb carries the needed information. In handling MWEs, it is also possible, and often needed, to add such new information that helps in achieving grammatically correct translation.

The rules for isolating MWEs are ordered so that the longer one wins. Problematic are such cases, where there are two contiguous MWE candidates with one or more shared members. This much discussed problem can be solved by adding context-based restrictions to rules for controlling rule application.

Although the different types of error-free CG rules are tedious to write manually, they can be produced with scripts from lexical lists.

5 Adding lexical glosses

The next step is to enrich each analysed word with the lexical representation of the target language. This is done so, that the POS category of the analysed word is found first, and then the lexical information is added from the lexicon of that POS category. In total, 13 POS categories are used in the system. Especially when translating from English, the identification of the correct POS category is important, because English is extremely ambiguous in this respect.

For making semantic disambiguation easier, the lexical glosses are ordered so that the most likely interpretation is the first one. There is no safe method for deciding which gloss should be considered as default, and often only thorough testing will help or statistical evidence can be used. Jörg Tiedemann has kindly helped in providing frequency lists produced using automatic word alignment (Östling and Tiedemann, 2016) on parallel corpora (mainly Europarl, Wikipedia headlines but also from OPUS[2]). Bilingual word lists are extracted from the aligned corpora and ranked by Dice scores and raw frequencies. Such words are discarded that include non-alphabetic characters and co-occurrence thresholds are used to further reduce the noise in the data. Separate lists are extracted for English multi-word-units that are aligned to single Finnish words and also for frequently aligned multi-word units on both sides. There is also a lemmatised version of the data. Lexical glosses are added in (4).

(4)

```
"<*he>" "he" { *hän Np9 FRONT , hänen , NO-
GLOSS , itse N8 FRONT } %SUBJ CAPINIT PRON
PERS NOM SG3
"<will>" "will" { NOGLOSS , aikoa V52-D ,
tulla V67 } %+FAUXV V AUXMOD
"<be>" "be" { olla V67b BE , eivät ole , ei
ole , NOGLOSS , joka Np13 , jotka Np14 } %-
FAUXV V INF
"<hanging_out>" "hang_out" { hengailla V67 ,
roikkua V52-A } %-FMAINV ING
"<on>" "on" { NOGLOSS M-ADE , NOGLOSS M-ILL ,
NOGLOSS M-PAR , NOGLOSS M-ELA , NOGLOSS M-ALL
, NOGLOSS M-ESS , NOGLOSS M-INE } %ADVL PREP
"<stages>" "stage" { vaihe N48 , lava N9 ,
näyttämö N2 FRONT } %<P N PL NOM
"<for_years_to_come>" "for_year_to_come" {
tulevina vuosina , tuleviksi vuosiksi } ADV
```

[2] Data available from http://opus.lingfil.uu.se and http://www.statmt.org/wmt16/translation-task.html

Because Finnish is a highly inflecting language, the lexicon needs precise instruction on inflection. Nouns and adjectives need a unique code for inflection in each case, in gradation, and in front/back concordance. Verbs have a large number of inflected forms, and also they follow gradation and front/back concordance rules. Not all of this need to be included into the transfer lexicon, but some anyway. For example, for transitive verbs it is useful to mark whether their preferred object case is partitive or accusative. Many of them use both, however, but in specific contexts. In addition to object argument, many verbs have also other arguments that require a certain case in inflection. Also such information should be added to the lexicon.

Compound words are common in Finnish, and their handling can be done in two places. The safest way is to handle them as MWEs. However, because compounding in Finnish is very productive, also more general methods should be provided. Compounds in Finnish are such that only the last member, the head, of the compound inflects. Therefore, it is possible to mark compound word candidates, which, if required contextual criteria are fulfilled, will be selected as first parts of the compound and later joined together with the second member. Even more than one member of compound words can thus be combined. This works with such English compounds that are composed of consecutive words without of-genitive structure.

For such words, for which there is no lexical gloss in the system, there is a default that the form in source language is copied as a gloss. It is given an inflection code according to the form of its last part.

Adding the lexical information is implemented using the Beta rewriting language.

6 Semantic disambiguation

Perhaps the most challenging phase in the current translation system is the semantic disambiguation. Much of the complexity comes directly from the source language English, the analysis of which does not offer many clues for performing semantic disambiguation. For example, English verbs do not mark whether they are transitive or intransitive, and the same verb functions in both roles. This creates a recurrent semantic disambiguation problem. The solution must be found on the basis of presence or absence of the object in the sentence. Also, the presence or absence of

the agent in passive sentences helps in disambiguation. The distinction between transitive and intransitive verbs is one of the few cases, where rather global rules can be written on semantic disambiguation. Most of the rules are on a low level, applying to relatively few cases.

An example of complicated rules is to identify whether the word *with* starts a relative clause or whether it is in some other role. The identification alone does not suffice. One should also know whether it should be translated with singular or plural form.

The rules on punctuation are different in English and Finnish. These differences can be handled as part of semantic disambiguation.

Such words that can occur as proper nouns and ordinary words are a problem in translation. A partial solution is that such words that have a capital initial letter and are not sentence-initial are likely to occur in both roles and are marked as proper name candidates. Then, on the basis of the strict list and environment, the candidates are selected as proper names. However, this method does not work, if the word is sentence-initial, because in this position all words start with a capital letter.

If the language analyser would add the so-called supersenses to the analysis, the rule writing would become easier. Such comprehensive supersense categories have been established for English (Schneider and Noah 2015; Hollenstein et al 2016). The current system makes use of such sense categories as TIME, PLACE, ANIMACITY, HUMANNESS, TRANSITIVITY etc.. However, these categories are not part of the analyser, but they are implemented in the transfer rule system. The further development of the system might reveal, that more clustering should be made.

When semantic disambiguation rules are applied, the rest of readings are handled so that the first interpretation is selected and the rest are removed. Except for a few specific cases, the system does not leave ambiguity to the readings (5).

(5)

```
"<*he>" "he" { *hän Np9 FRONT } %SUBJ CAPINIT
PRON PERS NOM SG3
"<will>" "will" { NOGLOSS } %+FAUXV V AUXMOD
"<be>" "be" { NOGLOSS } %-FAUXV V INF
"<hanging_out>" "hang_out" { hengailla V67 }
%-FMAINV ING
"<on>" "on" { NOGLOSS } M-ADE %ADVL PREP
"<stages>" "stage" { näyttämö N2 FRONT } %<P
N PL NOM
"<for_years_to_come>" "for_year_to_come" {
tulevina vuosina } ADV
```

7 Controlling singular and plural

One could expect that singular matches with singular and plural with plural in two languages. This is not the case, however. A typical case is that whereas English uses plural forms in nouns that have a number as a modifier, Finnish uses singular. Also adjective and pronoun modifiers in such structures are in singular.

The en-fdg parser does not mark the number in adjectives and some verb forms, which is why such tags must be controlled in great detail.

8 Adding inflection tags

Because Finnish is a highly inflecting language, and English is not, there is little such information inherited from the analysis of the source language that can be used in constructing the correct Finnish word forms. Therefore, such instructions must be added, mostly on the basis of the information added in lexical mapping.

Adding inflection tags takes place in two phases. First, the primary constituents of the sentence are tagged. Such constituents include the verb, the subject, the object, the indirect object, and various modifiers of the verb.

In the second phase, adjective, pronoun, and number modifiers are given inflection tags on the basis of the inflection tag given to the noun head in the first phase of tagging.

Because rule writing for such a complex network is prone to multiple simultaneous mappings, the rules are hierarchically ordered and reapplication is prevented. The rules are ordered according to approximate security, the most secure ones first and the least secure ones last. By the secure rule we mean the likelihood that the rule works correctly in all contexts. There is no strict dichotomy between secure rules and other rules. Rather there is a continuum. Added inflection tags, some redundant, are displayed in (6).

(6)

```
"<*he>" "he" { *hän Np9 FRONT } %SUBJ CAPINIT
PRON PERS NOM SG3
"<will>" "will" { NOGLOSS } %+FAUXV V AUXMOD
SG PRES
"<be>" "be" { NOGLOSS } %-FAUXV V INF SG PRES
"<hanging_out>" "hang_out" { hengailla V67 }
%-FMAINV ING SG PRES
"<on>" "on" { NOGLOSS } M-ADE %ADVL PREP
"<stages>" "stage" { näyttämö N2 FRONT } %<P
N PL NOM ADE
"<for_years_to_come>" "for_year_to_come" {
tulevina vuosina } ADV
```

9 Marking stem boundary

There are 107 different inflection classes for Finnish nominals and verbs. The list of Kotimaisten Kielten Tutkimuskeskus (Research Centre of National Languages) has fewer categories, but they are insufficient for describing all word types. The number of inflection categories could certainly be reduced by applying a finite state machine for controlling part of variation.

There was no linguistic theory behind selecting the stem boundary marking. The solution was purely practical. The boundary mark was put to the point, which made it possible to produce all inflected forms of the word type.

10 Converting inflection tags to surface forms

Each inflection tag is converted to near-surface form using Beta-rules. The system first checks the inflection code, marks it as checked, and then looks for the other codes of that inflection class. If the path leads successfully to a surface form suffix, it is added after the suffix tag.

The process is not simple, however, because the reading may have two or three inflecting words, and each must be given the correct inflection. The danger of mixing the suffixes is avoided by joining the found suffix immediately to the word. A second, and possibly third, round is then run for finding the correct suffixes for the rest of words in that reading.

The suffixes are joined to the whole lexical word and not directly to the stem. This is done, because sometimes the correct front form can be decided only when the final part of the lexical word is present. Note that the inflection suffixes are not necessarily final. By default, suffixes are given the back vowel treatment (7). If the word requires front vowel treatment, conversion rules modify the suffix accordingly (8).

(7)

```
"<*he>" "he" { *h:än :Np9 FRONT } %SUBJ
CAPINIT PRON PERS NOM SG3
"<will>" "will" { NOGLOSS } %+FAUXV V AUXMOD
PRES SG
"<be>" "be" { NOGLOSS } %-FAUXV V INF PRES SG
"<hanging_out>" "hang_out" { hengail:la+ee
:V67 } %-FMAINV ING PRES SG
"<on>" "on" { NOGLOSS } M-ADE %ADVL PREP
"<stages>" "stage" { näyttämö:+illa :N2 FRONT
} %<P N PL ADE
"<for_years_to_come>"   "for_year_to_come"   {
tulevina vuosina } ADV
```

11 Front/back concordance

The decision on whether the ending of the word gets a front or back vowel treatment is done on the basis of the vowel structure of the word. In addition to this, the end part of the lexical word may affect the precise surface form of the word. For example, the rule may require that the back vowels of the suffix must be converted to corresponding front vowels. This conversion is not always one-to-one process, because a back vowel may have two corresponding front forms. This can be decided on the basis of the last vowel of the lexical word. Therefore the full lexical word must be present when conversion rules are applied. Front vowel conversion is displayed in (8).

(8)

```
"<*he>" "he" { *h:än } %SUBJ CAPINIT PRON
PERS NOM SG3
"<will>" "will" { NOGLOSS } %+FAUXV V AUXMOD
PRES SG
"<be>" "be" { NOGLOSS } %-FAUXV V INF PRES SG
"<hanging_out>" "hang_out" { hengail+ee } %-
FMAINV ING PRES SG
"<on>" "on" { NOGLOSS } M-ADE %ADVL PREP
"<stages>" "stage" { näyttämö:+illä } %<P N
PL ADE
"<for_years_to_come>"   "for_year_to_come"   {
tulevina vuosina } ADV
```

12 Controlling word order

POS tags are the most important keys in controlling the word order in target language. In the current language pair, the most important features that require word reordering are the prepositions and the of-genitive. Finnish most often uses postpositions, and it does not have equivalent for of-genitive. Also, passive structures with agent are missing in Finnish, which causes complex changes in word order.

We have written the reordering rules with Perl. In order to simplify rule-writing, we have moved the POS tag to the beginning of the reading of each word and changed the whole input into sentence-per-line format. Using this format, it is fairly easy to write new reordering rules. For each word type applies the same description, and only the POS tag changes. In case additional information is needed, it is available in the description of the word.

13 Discussion

The current translation system tries to make maximal use of the lexicon and grammar of source and target languages. A sentence in

source language is converted through subsequent phases into target language. No purely statistical choices are used. In order to reduce unnecessary rule writing, defaults are used where feasible.

Such rules that need contextual control for their application are implemented using the CG3 environment. Such cases are, apart from the parsing component of English, the correction module for the output of the parser, the isolation and treatment of MWEs, the semantic disambiguation, the control of singular and plural forms, and the modules for adding primary and secondary tags for facilitating inflection. The rest of rules are implemented using rewriting rules in Beta or Perl, whichever is feasible in each case.

The periodic development with this language pair was started in 2015, using IT and medical domains as test environments. The work with news texts started in March 2017, and the work with this domain is just in the beginning. Especially the vocabulary of the domain is very defective, and also the isolation of MWEs needs much work.

Our own estimation of the feasibility of the rule-based approach to the current task is that the more grammatical the sentences are, the better the result. The ordinary news reporting can be translated satisfactorily, but sport news and other types of less grammatical texts are a big problem.

References

Barreiro Anabela, Bernard Scott, Walter Kasper and Bernd Kiefer, 2011. OpenLogos Rule-Based Machine Translation: Philosophy, Model, Resources and Customization. *Machine Translation*, volume 25 number 2, Pages 107-126, Springer, Heidelberg, 2011. ISSN 0922-6567, doi:10.1007/s10590-011-9091-z

Bick Eckhard and Tino Didriksen, 2015. CG-3 - Beyond Classical Constraint Grammar. In: Beáta Megyesi: *Proceedings of NODALIDA 2015*, May 11-13, 2015, Vilnius, Lithuania. pp. 31-39. Linköping: LiU Electronic Press. ISBN 978-91-7519-098-3.

Forcada Mikel L., 2006. Open-source machine translation: an opportunity for minor languages in B. Williams (ed.): *Proceedings of the Workshop "Strategies for developing machine translation for minority languages* (5th SALTMIL workshop on Minority Languages)" (organised in conjunction with LREC 2006 (22-28.05.2006)). Genoa, Italy, pp. 1-6.

Hollenstein Nora; Nathan Schneider and Bonnie Webber, 2016. Inconsistency Detection in Semantic Annotation. In *Proceedings of LREC-2016*. Pp. 3986-3990.

Hurskainen A., 1992. A Two-Level Computer Formalism for the Analysis of Bantu Morphology. An Application to Swahili. *Nordic Journal of African Studies,* 1(1), 87-122.

Hurskainen A., 1996. Disambiguation of morphological analysis in Bantu languages. In *Proceedings of the 16th conference on Computational Linguistics*. Copenhagen:ACL. Vol.1, pp.568-573.

Hurskainen A., 2004. Optimizing disambiguation in Swahili. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, Geneva, Switzerland, 23-27 August 2004, pp. 254–260. Genoa: [International Conference on Computational Linguistics].

Hurskainen A., 2006. Constraint Grammar in Unconventional Use: Handling complex Swahili idioms and proverbs. In: Suominen, Mickael et.al. (ed.) *A Man of Measure: Festschrift in Honour of Fred Karlsson on his 60th Birthday*. Special Supplement to SKY Jounal of Linguistics, Vol. 19 (ISSN 1796-279X), pp. 397-406. Turku: The Linguistic Association of Finland.

Hurskainen A., 2007. A rule-based environment for Swahili development. *MultiLingual,* 18(8), 53-58. ISSN 1523-0309.

Hurskainen A., 2012. Quality Swahili machine translation. *MultiLingual,* 23(7), 39-42. ISSN 1523-0309.

Järvinen Timo and Tapanainen Pasi, 1997. A Dependency Parser for English. *Technical Reports*, No. TR-1. Department of General Linguistics, University of Helsinki.

Karlsson Fred, 1990. Constraint grammar as a framework for parsing running text. In: Karlgren Hans (ed.), *Proceedings of 13th International Conference on Computational Linguistics*, volume 3, pp. 168-173, Helsinki, Finland.

Karlsson Fred, 1995. Designing a parser for unrestricted text. In: Karlsson F. et al (Eds), *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text,* 1-40. Berlin: Mouton de Gryuter.

Karlsson Fred; Atro Voutilainen, Juha Heikkilä and Arto Anttila, 1995. *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text.* Mouton the Gruyter.

Pirinen Tommi A., 2015. Omorfi —free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics NODALIDA 2015*, pages 313–315.

Ranta Aarne*, 2011.* Grammatical Framework. Programming with Multilingual Grammars*. CSLI Publications, Center for the Study of Language and Information.* pp. 8–9. ISBN 978-1-57586-627-7.

Scott B. and Barreiro A. 2009. OpenLogos MT and the SAL representation language. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation* / Edited by Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Francis M. Tyers. Alicante, Spain: Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos. 2–3 November 2009, pp. 19–26.

Schneider Nathan and Noah A. Smith, 2015. A Corpus and Model Integrating Multiword Expressions and Supersenses. In *Proceedings of NAACL-HLT-2015*. Pp. 1537-1547

Silberztein M., 2015. *La formalisation des langues : l'approche de NooJ.* ISTE: London (426 p.).

Tapanainen Pasi, 1996. *The Constraint Grammar Parser CG-2.* University of Helsinki Publications No. 27.

Tapanainen Pasi, 1999. *Parsing in Two Frameworks: Finite-state and functional dependency grammar.* Ph.D. Dissertation, Department of General Linguistics, University of Helsinki.

Tiedemann Jörg, Fabienne Cap, Jenna Kanerva, Filip Ginter, Sara Stymne, Robert Östling and Marion Weller-Di Marco, 2016. *Phrase-Based SMT for Finnish with More Data, Better Models and Alternative Alignment and Translation Tools.* WMT 2016: 391-398.

Östling Robert and Jörg Tiedemann: Efficient Word Alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics (PBML),* Number 106, pp. 125–146.