A Consensus Approach for Annotation Projection in an Advanced Dialog Context

Simon Julien[1], Philippe Langlais[2] and Réal Tremblay[3]

[1] DIRO, Université de Montréal juliensi@iro.umontreal.ca
[2] DIRO, Université de Montréal felipe@iro.umontreal.ca
[3] Nuance Communications real.tremblay@nuance.com

Abstract. Data annotation is a common way to improve the reliability of advanced dialog applications. Unfortunately, since those annotations are highly language-dependent, the universalization can become a very lenghty process. Even though some projection methods exist, most of them require a deeper level of annotation than the one used for advanced dialogs. In this paper, we present a consensus approach that exploits the specificities of a sparse annotation in order to do the data projection.

1 Introduction

Conversing with a computer can be tedious, which is why historically, many constraints were imposed on the dialog to ensure a viable (yet not very natural) conversation: short answers, step-by-step value collection, systematic confirmation, etc. Progress in computer technology lifted most of those constraints and, with current "advanced dialog" applications, it is now possible to use longer sentences, along with much more complex dialog patterns.

A common way to create such an application, or to enhance one, is to annotate a considerable amount of data. The annotations provide the desired parse results for typical sentences, from which new parses can be deducted, and then, from all the annotated data, a parsing grammar can be automatically generated. A major drawback of this otherwise easy method is to increase the development time, specially when it comes to universalization: all annotations are language-dependent, so the work done for a first language (say English), needs to be renewed for any further language (e.g. French, Spanish, Russian, Chinese...).

While many resource projection methods exist (Hwa et al., 2004; Santaholma, 2008; Kim et al., 2010; Bouillon et al., 2006), none of them are fit to be used directly on the characteristic sparse annotation of advanced dialogs. For example, (Kim et al., 2003) uses a full-tree annotation hierarchy for its multilingual grammar development. Similarly, (Santaholma, 2005) needs access to the grammar syntax of each sentence to perform its speech-to-speech translation. That syntax has to be pinpointed in some way, usually through extra annotation. In all cases, the detailed annotation creates a deep level of abstraction that can be exploited by the proposed algorithms for the resource projection.

Unfortunately, meta-information is too sparse when it comes to advanced dialogs, where most of the sentence is left unannotated (even though the whole

sentence is used to determine the context). For instance, in "I would like to go to Newark, not New York", the annotation would typically be targeted on "Newark" and "New York", while "I would like to go to" and "not" would be left unannotated (their modifying action would however be accounted for in the way "Newark" and "New York" are annotated).

In this paper, we present a consensus method that projects a sparse annotation from a source language (English) to a target one (French), using Moses (Koehn et al., 2007), a public Statistical Machine Translation system (SMT – Koehn et al., 2003 ; Chiang, 2005). That method is directly aimed towards advanced dialog applications. We also provide a quantitative evaluation of the method against other intuitive solutions, using real data from two domains (Airline and Insurance). The datasets are described in Section 2, while the different methods are presented in Section 3. The comparative analysis is then conducted in Section 4, just before the conclusion of Section 5.

2 Data and Protocol

Prior to the deployment of an application, lots of representative utterances are typically gathered and manually annotated. The number of utterances varies according to each domain, but it is usually no less than a few thousands. Then, from those tagged utterances, a semantic parser is trained and can be used to parse new and unseen utterances. Our goal is to alleviate the annotation process for any further language, using previous annotation in the source language and machine translation. Since manual annotation is very time-consuming, this task is a real concern and a real challenge.

2.1 Datasets

In this work, we considered two datasets: the first one comes from a booking application in the airline domain, and the second one comes from a more general "how may i help you" application in insurance. English utterances have been internally collected at Nuance (duplicates were removed) and randomly split into training and testing sets, before they were manually translated in French. French duplicates, if any, were also removed. That way, we ensured that the translation of any source utterance in the train set is not in the target test set, where it could artificially boost the performance metrics. Typical examples from one of the application we studied are reported in Table 1.

For ease of understanding, we must define and illustrate the notions of tag, bounding tag and pattern:

Definition 1. *A* tag *is a conceptual entity that identifies the semantical role of contiguous words in a sentence. A* tag *can also identify the semantical role of contiguous tags, or of a mix of words and tags.*

Table 1. Examples of utterances from the Airline application, along with their tags and patterns.

utterance	tags	pattern
Monday, I'm taking a plane bound for Christmas Island	**DAY_OF_WEEK**(Monday) **DATE**(DAY_OF_WEEK) **DEPARTURE_DATE**(DATE) **TERRITORY**(Christmas Island) **LOCATION**(TERRITORY) **DESTINATION**(LOCATION)	DEPARTURE_DATE, I'm taking a plane bound for DESTINATION
for Christmas, I need to take a flight between Melbourne and Sydney	**HOLIDAY**(Christmas) **DATE**(HOLIDAY) **DEPARTURE_DATE**(DATE) **CITY**(Melbourne) **LOCATION**(CITY) **ORIGIN**(LOCATION) **CITY**(Sydney) **LOCATION**(CITY) **DESTINATION**(LOCATION)	for DEPARTURE_DATE, I need to take a flight between ORIGIN and DESTINATION

Definition 2. *A* pattern *is a utterance in which all tagged words are replaced by their more general bounding tag. If a tag encompasses more than one word, a single instance of the tag replaces all the words.*

Take, for example, the utterance "leaving June 6^{th}". A tag MONTH would target the literal "June", a tag DAY would target "6^{th}" and a tag DATE would target those two previous tags (MONTH and DAY). Also, because of the context introduced by the keyword "leaving", a final tag DEPARTURE_DATE would encompass the lower tag DATE, in order to identify more throughly the semantical role of the underlying words. The DEPARTURE_DATE tag, which is not part of any other tag is considered the most general one. A common way to illustrate all those relations is with the use of a tree: "leaving DEPARTURE_DATE(DATE(MONTH(June) DAY(6th)))". As for Definition 2, the pattern of the sentences "leaving June 6^{th}" and "leaving tomorrow" would likely be the same: "leaving DEPARTURE_DATE".

The notion of pattern is important, because the annotation (along with a list of recognized literals) is often sufficient to parse very close utterances. For instance, it's easy to parse "somewhere in Honduras" once you've seen "somewhere in Belgium". Furthermore, that pattern redundancy will later be used in our consensus projection method.

The main characteristics of the two datasets we gathered are reported in Table 2 and 3 along with the number of different patterns and the percentage of the test patterns not seen at training time (OOVP%).

The number of utterances in both tasks is roughly similar (about 2500 utterances, two thirds of which are being used for training). We also observe that the tagset of the airline application is much larger than the one for insurance. This is because insurance data is more action driven (file a claim, deposit, withdraw, transfer ...), while airline data is more about targeting concepts (origin, destination, flight class ...). While distinct tags are needed for the latter type of data, an action usually only modifies existing tags, without creating new ones (for example, an amount becomes a deposit). This is also why the number of patterns and the OOVP% are more considerable in the insurance domain. Since more words are left unannotated, more different patterns arise, and with more different patterns to see, more are still unseen when it comes to testing.

Table 2. Data from the Airline domain

	English			**French**		
	total	train	test	total	train	test
Utterances	2449	1835	614	2233	1693	540
Tags	4163	3140	1023	3849	2938	911
Patterns	910	733	289	969	767	285
OOVP%			29.8%			38.2%

Table 3. Data from the Insurance domain

	English			**French**		
	total	train	test	total	train	test
Utterances	2459	1844	615	2313	1715	598
Tags	2663	2011	652	2426	1793	633
Patterns	1874	1454	538	1891	1459	551
OOVP%			69.3%			70.4%

2.2 Statistical Machine Translation

As for the bitexts used to train the SMT engine, we willingly chose them out-of-domain. Even if in-domain training provides better results (Koehn and Schroeder, 2007), the existence of in-domain bitexts is unsure in a real-life situation. Therefore, we used public corpora that anyone could find, from official records of the Canadian Parliament[4] (Hansard) and from movie subtitles[5] (OpenSubtitles2011 – Tiedemann, 2009). The main characteristics of the bitexts we used are reported in Table 4.

[4] http://www.isi.edu/natural-language/download/hansard/
[5] http://opus.lingfil.uu.se

Table 4. Corpora used to train the SMT engine

corpus	docs	phrases	tokens (en)	tokens (fr)
OpenSubtitles2011 (fr-ca)	24116	19.7×10^6	119.0×10^6	114.5×10^6
Hansard	2	1.2×10^6	19.8×10^6	21.2×10^6

2.3 Evaluation

The evaluation is pretty straightforward: for all the utterances of the test set, the retrieved tags are compared to the expected ones, using precision, recall and F_1 score (micro-averaged). The tags of a given utterance are directly inferred from the annotated sentences of the training set, with possibly some inner-filler words (that is, words between two recognized patterns that can be ignored). Basically, each tagged utterance becomes a recognized parsing rule, which is directly applied if possible (longer rules are preferred over shorter ones). Better parsing methods exist, but this simplier approach is useful to quickly zero in a better annotation projection algorithm.

3 Experiments

A short summary may be helpful here: what we are trying to do is to use n tagged utterances in a source language (English) to generate automatically, via SMT, m tagged utterances in a target language (French). Those m utterances, along with t "true" tagged target utterances, are then used to parse test sentences, from which precision, recall and F_1 are calculated. Given that the t "true" utterances from the French train set are never altered in any way, an algorithm that translates the n source utterances into the m ones will be considered better than another one if its F_1 score is higher. Below, we present three of those algorithms.

3.1 Baseline

For the baseline experiment, no SMT is used at all (all source utterances are ignored, i.e. $m = 0$). Instead, only samples from the French train set are randomly selected and directly used to train the semantic parser for the target language. This is a way to draw the learning curve of the applications over time, when $t = \{0, 50, 100, 150, \ldots\}$ samples are available in the current language, regardless of any other language.

3.2 Translation and Annotation

In this method, all source utterances are translated using the SMT. The literal expression encompassed by each tag is also translated. Then, the global annotation of a translation is inferred by searching each local literal translation, on which the original tag is restored. See Figure 1 for a simple example.

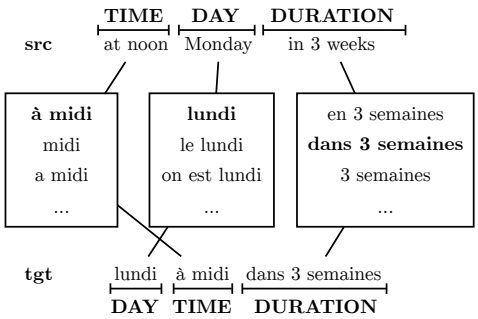

Fig. 1. Annotation projection through local search

Usually, more than one translation candidate is returned (but they are ranked according to their SMT score). Therefore, only the first candidate for which all tags were successfully restored is retained. For instance, in Figure 1, if "lundi dans 3 semaines" would have preceded "lundi à midi dans 3 semaines", the latter would still have be retained, due to the missing TIME tag ("à midi") in the former. Likewise, a tag will be considered missing if none of its 3 first translation candidates could be found. Even with those restrictions, there is almost always an adequate translation candidate for each source utterance, so $m \approx n$ ($m \leq n$) with this method.

This annotation projection method was preferred over a more intuitive approach, where the annotation alignment is combined with the translation alignment to retrieve a given target annotation. This is mainly because ghost concepts (see Figure 2) are sometimes wrongfully retrieved, a problem avoided with the projection through local search.

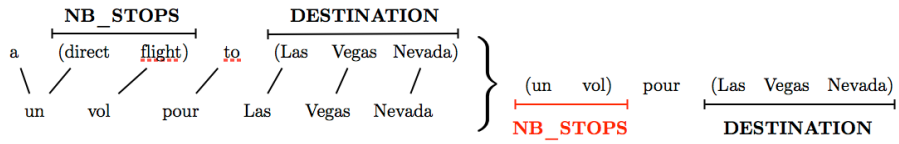

Fig. 2. Erroneous annotation projection when a local concept is missing from a translation

3.3 Consensus Approach

The consensus approach we present uses the same algorithm to project annotation between a source utterance and its translation, except it first focuses on

identifying good translation candidates. In order to do so, source examples are no longer considered individually, but in clusters, using patterns as classifiers. For instance, the cluster "from ORIGIN to DESTINATION" would comprise utterances such as "from Toronto to Denver", "from Calgary Alberta to France", "from Italy to somewhere in Europe" and so on.

Because the SMT computes its transfer probabilities on each token individually, two utterances with the same source pattern will not necessarily produce the same n-best list of translation patterns. In fact, because a same concept (e.g. DATE) can be expressed in numerous ways ("today", "next weekend", "June 18$^{\text{th}}$", "on Christmas", ...), significative differences arise in the translation candidates of yet very-close source utterances. On that matter, the example of Table 5 is very representative.

Table 5. Various n-best translation patterns from the source pattern "I'd like to go from ORIGIN to DESTINATION"

src utterance	n-best translation patterns	
I'd like to go from **Boston** to **New York**	je voudrais aller **à** ORIGIN **à** DESTINATION	✗
	je voudrais aller **de** ORIGIN **à** DESTINATION	✓
	voudrais aller **de** ORIGIN **à** DESTINATION	✓
	je voudrais aller **à** ORIGIN **pour** DESTINATION	✗
I'd like to go from **Montreal** to **New York**	je voudrais aller **de** ORIGIN **à** DESTINATION	✓
	je voudrais aller **à** ORIGIN **pour** DESTINATION	✗
	je voudrais **y** aller de ORIGIN DESTINATION	✗
	je voudrais aller **à** ORIGIN **de** DESTINATION	✗
I'd like to go from **Chicago Illinois** to **London UK**	je voudrais aller **de** ORIGIN **en** DESTINATION	✗
	je voudrais aller **à** ORIGIN **de** DESTINATION	✗
	je voudrais aller **de** ORIGIN **pour** DESTINATION	✓
	je voudrais aller **de** ORIGIN **à** DESTINATION	✓

Given those differences, our hypothesis is that recurrent translation patterns must be more reliable than unfrequent ones. A simple poll-voting algorithm with the 10 first translation candidates proved us right, at least on our datasets. And because the SMT is not completely wrong after all, a more effective way to pinpoint good translation patterns is to weight each one according to their translation rank. An inversely linear relationship turned out to be the best option.

Therefore, our consensus approach works as follows: each source utterance is sorted according to its pattern. Then, for each cluster, the best translation patterns are determined through a weighted-polling method. The 10-best translation candidates are considered for each utterance, then the score of each associated translation pattern is incremented according to the translation rank.

For example, the first example of Table 5 would increase the score of "je voudrais aller **à** ORIGIN **à** DESTINATION" by 10, the score of "je voudrais aller **de** ORIGIN **à** DESTINATION" by 9, "voudrais aller **de** ORIGIN **à** DESTINATION" by 8, etc. Once each example from a cluster voted, the 2 translation patterns with the highest scores are retained and considered "reliable". Finally, only pairs [src utt ; translation] in which the translation conforms to a reliable pattern are used to become the m tagged utterances.

Our consensus approach shares similarities with one of (Bangalore et al., 2002) with two important differences. First, we do not combine the translations of various off-the-shelf translation engines, but use a single system that we fully trained. We believe this is an easier setting to deploy, but further investigations are needed to compare the two approaches. Also, we directly use the annotation available in the source language, making our approach better tailored to our needs.

4 Results and Analysis

The three methods have been implemented and tested according to a growing number of real target utterances (examples from the French train set, which were manually annotated). For robustness purposes, the results shown in Figure 3 and 4 correspond to the average result over 5 repetitions (new manually-annotated samples were chosen at random each time).

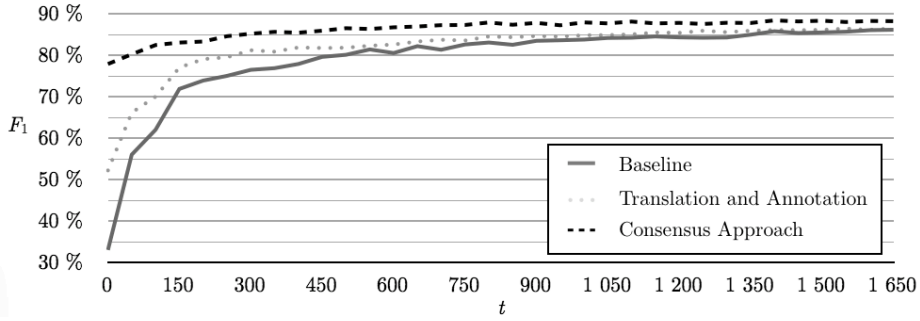

Fig. 3. F_1 variation on the Airline test set by the nb of first-hand utterances (t)

A first noticeable thing is that the F_1 baseline curves do not start at 0%, but near 30%. That is because roughly 30% of the tags in each test set can be identified with minimal knowledge. Those tags are usually basic concepts that can be listed[6] (e.g. list of countries). All other tags (±70%) are modified by

[6] main exceptions being DATE and TIME concepts, which cannot be listed per se, but are frequent enough to get a special treatment

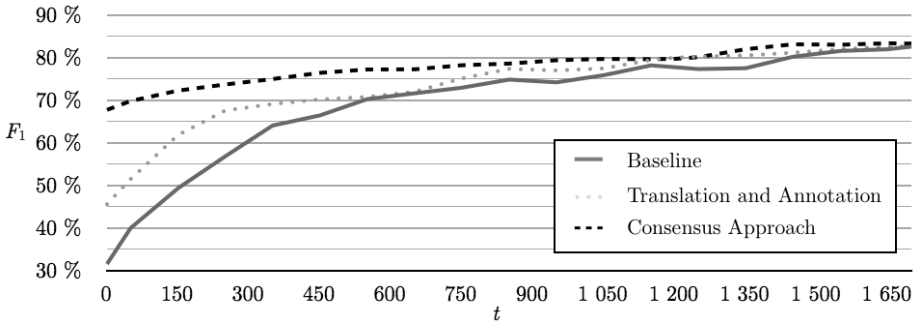

Fig. 4. F_1 variation on the Insurance test set by the nb of first-hand utterances (t)

the context (e.g. a COUNTRY that becomes a DESTINATION in "going to Finland").

The consensus approach curve clearly stands out, whether in the Airline or the Insurance domain. Furthermore, the performance of our method when only a few target annotated examples are being used is very close to the approach obtained when all target examples are considered, which means the projected utterances are pretty good estimations of the real data. Therefore, few real target examples bring patterns that were not deducted from the original language, and few projected patterns are wrong (which is arguably the main problem of the "Translation and Annotation" approach). Because more than one target pattern can be kept for each source pattern, the consensus approach even keeps its edge over the baseline when all the Airline training data is considered (see Figure 3). A higher OOVP% explains why the approach scored lower at $t = 0$ on the Insurance test set than on the Airline test set.

4.1 Cardinality and Efficiency

Given that the consensus approach relies on various neighbour utterances, it is tempting to artificially increase the number of examples per cluster. One easy way to do so is to exploit the source annotation. That annotation is hierarchical, so from a common annotation node, new examples can be automatically generated. For instance, from the example "my departure date is on the 4^{th} of July", where the annotation looks like "my departure date is on DEPARTURE_DATE(DATE(the DAY(4^{th}) of MONTH(July)))", the subconcept DATE can easily be replaced by any other encountered DATE, creating new tagged examples for the cluster "my departure date is on DEPARTURE_DATE".

Over-generating data is promising, although real-life experiments in our domains showed very shy improvements. More importantly, over-generation gave us a way to measure how the clusters' cardinality impacted the results through our consensus approach. In order to do so, over-generation was used to ensure each source pattern had at least 4 associated examples. Then, the consensus

approach was repeatedly used, always considering all the available source patterns, but with more and more underlying examples (first, a single example was used per pattern, then 2 examples per pattern, then 3 and then 4). Once again, experiments where run 5 times to ensure a minimal robustness.

Table 6. Impact of the clusters' cardinality

	Airline			**Insurance**		
cardinality	P	R	F_1	P	R	F_1
$n = 1$	44.1%	51.8%	47.6%	37.4%	43.6%	40.2%
$n = 2$	51.2%	56.5%	53.7%	42.1%	48.2%	44.9%
$n = 3$	56.0%	60.8%	58.3%	50.4%	55.0%	52.6%
$n = 4$	59.5%	63.2%	61.3%	54.3%	56.0%	55.2%

As Table 6 shows, the results get better as the clusters' cardinality increases. This seems to confirm that over-generation could be useful in cases where meaningful clusters are under-represented in the training set. It also confirms that the average translation of close utterances is more reliable than each utterance considered alone, at least in the cases of interest. Over-generation was not used in the results of Figure 3 and 4, where each clusters had a various number of associated examples.

4.2 Proximity with the SMT

At this point, another reasonable hypothesis would be that source examples "close" to the SMT probably produce overall better translation candidates. In other words, instead of mixing various translation candidates through our consensus approach, maybe it would be more efficient to identify a single source utterance per pattern, that source utterance being the one which is the most familiar to the SMT. After all, it is the SMT that performs the translation. So if, for example, "Munich" is much more frequent in the training bitexts than "Albuquerque", we should replace the latter with the former everywhere.

In order to test this intuition, we trained a bigram language model on the source side of the SMT training bitexts. We were then able to compute a proximity score on the training data of our two domains. The best candidate for each cluster was retained alone, and used to do the data projection (through the Translation and Annotation approach). The following results were obtained.

A small increase can be observed over the "Translation and Annotation" method, but the results are far from the ones obtained with the consensus approach. It would seem that examples close to the SMT are good translation candidates afterall, but not as good as an average translation over more examples. That is mainly because the SMT itself is not trained with data relevant to the current domain (in our case, airline or insurance), but with general data from various sources (in our case, parliament debates and movie subtitles). Therefore,

Table 7. Impact of the proximity with the SMT

	Airline			Insurance		
method	P	R	F_1	P	R	F_1
Consensus	74.9%	81.1%	77.9%	64.3%	70.2%	67.1%
Transl & Annot	49.5%	53.8%	51.6%	43.8%	44.7%	44.2%
SMT	52.6%	57.1%	54.7%	47.0%	49.0%	48.0%

an example that is close to the SMT will most likely be translated as if it came from a debate or a movie, rather than from an airline or an insurance application. Still, an efficient weighting of examples prior to their translation appears like a promising way to improve our current algorithm.

5 Conclusion

From our experience, the consensus approach is an easy and effective way to project data for advanced dialog purposes. However, the reader must bear in mind that two main limitations could arisen if our method were to be used in another context. First, with the current algorithm, a source pattern can have no target pattern at all, should some concepts be completely absent from each translation candidates. In some (rare) cases, clusters can be ignored entirely. Also, the algorithm widely exploits the fact that wrong translations are often harmless in the current task, because no user will ever uses them in a real-life situation. Therefore, it is a minor error to deduct a rule from a completely wrong translation (it will simply become a dead rule in the grammar), but it is a major one to deduct a false rule from a meaningful translation.

Nonetheless, it appears that the projection of annotations in an advanced dialog context, even without a dedicated translation system, is perfectly feasible. In this regard, the consensus approach we presented is very effective. The projection is done automatically, and no further tagging is required. In the future, an automatic way to weight each examples according to their prior reliability could be helpful. Our method has also yet to be tested on a bigger scale. From a more general point of view, we believe that any algorithm meant to address this specific problem of data projection should exploit the redundancy of the source annotation, and our consensus approach is an easy and viable solution to do so.

References

[1] BANGALORE, Srinivas; MURDOCK, Vanessa; RICCARDI, Giuseppe (2002) Bootstrapping Bilingual Data Using Consensus Translation for a Multilingual Instant Messaging System, In: *Proceedings of the 19th International Conference on Computational Linguistics*, p. 1–7.

[2] BOUILLON, Pierrette; RAYNER, Manny; NOVELLAS, Bruna; NAKAO, Yukie; SANTAHOLMA, Marianne; STARLANDER, Marianne; CHATZICHRISAFIS,

Nikos (2006) Une grammaire multilingue partagée pour la traduction automatique de la parole, In: *Proceedings of Traitement Automatique des Langues Naturelles*, p. 155-173.

[3] CHIANG, David (2005) A Hierarchical Phrase-Based Model for Statistical Machine Translation, In: *ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, p. 263-270.

[4] HWA, Rebecca; RESNIK, Philip; WEINBERG, Amy; CABEZAS, Clara; KOLAK, Okan (2004) Bootstrapping Parsers via Syntactic Projection across Parallel Texts, In: *Natural Language Engineering, 11(3)*, p. 311-325.

[5] KIM, Roger; DALRYMPLE, Mary; KAPLAN, Ronald; KING, Tracy H.; MASUICHI, Hiroshi; OHKUMA, Tomoko (2003) Multilingual Grammar Development via Grammar Porting, In: *Proceedings of the ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*, p. 49-56.

[6] KIM, Roger; JEONG, Minwoo; LEE, Jonghoon; LEE, Gary Geunbae (2010) A Cross-lingual Annotation Projection Approach for Relation Detection, In: *Proceedings of the 23rd International Conference on Computational Linguistics*, p. 564-571.

[7] KOEHN, Philipp; OCH, Franz Joseph; MARCU, Daniel (2003) Statistical Phrase-Based Translation, In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, p. 48-54.

[8] KOEHN, Philipp; SCHROEDER, Josh (2007) Experiments in Domain Adaptation for Statistical Machine Translation, In: *StatMT '07 Proceedings of the Second Workshop on Statistical Machine Translation*, p. 224-227.

[9] KOEHN, Philipp; HOANG, Hieu; BIRCH, Alexandra; CALLISON-BURCH, Chris; FEDERICO, Marcello; BERTOLDI, Nicola; COWAN, Brooke; SHEN, Wade; MORAN, Christine; ZENS, Richard; DYER, Chris; BOJAR, Ondrej; CONSTANTIN, Alexandra; HERBST, Evan (2007) Moses: Open Source Toolkit for Statistical Machine Translation, In: *Annual Meeting of the Association for Computational Linguistics*, p. 177-180.

[10] SANTAHOLMA, Marianne (2005) Linguistic Representation of Finnish in a Limited Domain Speech-to-Speech Translation System, In: *Proceedings of the 10th Conference on European Association of Machine Translation*, p. 226-234.

[11] SANTAHOLMA, Marianne (2008) Multilingual Grammar Resources in Multilingual Application Development, In: *Proceedings of Grammar Engineering Across Frameworks Workshop*, p. 25-32.

[12] TIEDEMANN, Jörg (2009) News from OPUS – A Collection of Multilingual Parallel Corpora with Tools and Interface, In: *Recent Advances in Natural Language Processing volume V*, p. 237-248.