Example-Based Machine Translation of Short Phrases Using the Context Equivalence Principle

Konstantin Tretyakov

Abstract

In this paper we consider machine translation in a very specific setting of *computer-assisted software localization*. This setting imposes certain requirements on the kind of translation assistance expected from the computer, and we believe that simple example-based techniques suit best for the task. We propose a very simplistic method, that exploits the assumption of context-equivalence common to other example-based techniques. The novelty of the approach (as we believe) lies in the fact that it uses *no other* linguistic heuristics such as word-wise alignments. As a result, the method produces very reliable translations at the price of not being able to translate all sentences – the behaviour that is well-suited for the task at hand. We have tested the method on a realistic dataset of localized messages and labels from a typical web-application, as well as a dataset of messages from KDE games. In different tests the method succeeded in perfectly translating between 5 and 37 percent of the messages with a negligible amount of mistakes. We find this result quite successful.

1 Introduction

Machine translation (MT) is perhaps the single most important application of computational linguistics, with a wide range of methods being developed and applied in a variety of contexts [Mitkov, 2003]. In general, the task of a machine translation system is to translate text from one language into another with or without human assistance. A distinction should be made between systems where MT is used to help people *understand* foreign text, and where it helps to *produce translations*. Whereas in the former case the quality of translation does not matter too much, as long as the meaning is preserved, in the latter setting of so-called *computer-assisted translation* systems, quality is the main concern. It is this latter kind of MT that we are dealing with in this paper.

Although significant progress has been achieved in the former type of MT during recent decades, mostly thanks to the development of *statistical machine translation* methods [StatMT], the technology behind *computer-assisted translation* (CAT), it seems, has yet to reach its critical point. We cannot yet trust computers to produce high-quality translations, and most of the contemporary CAT systems focus foremost on providing a convenient user-interface for the translator (easy access to online dictionaries, remote terminology databanks, stores of previously translated text, text highlighting and spelling correction, etc) than on the actual *translation* per se. The reasons for that tendency are probably quite natural. In order for an automated translation method to be useful in a CAT system, it has to produce *reliable* translations, that *exactly* correspond to what a human translator would produce. Having a computer assistant that requires the translator to do significant corrections to its suggestions is nearly as good as having no assistant at all. Therefore, in order to produce reliable translations, the system should quite strictly follow the rules and style of the human translator – a task often inherently

complicated for a statistical MT system, and requiring a lot of tedious customized rule-writing for a rule-based system.

The philosophy of *example-based machine translation* (EBMT) [Nagao, 1984; Turcato and Popowich, 2001] combines the features of rule-based and statistical approaches in a manner that seems favorable for the task at hand. EBMT, however, has been reported to require quite large corpora of aligned sentences for reasonable performance [Brown, 1996]. Also, typical implementations of the EBMT approach [Nagao, 1984; Kaji et al., 1992; Veale and Way, 1997; Brown, 1996, 2001] seem to exploit quite a lot of ad-hoc heuristics and statistical reasoning, which puts them closer to their statistical cousins, and thus diminishes the potential reliability of the translations. In this work we counter the two named issues by designing an EBMT system based on a tightly controlled set of simple heuristics (a single principle, in fact), and apply it for a very limited language consisting of short messages.

2 Motivation

The original motivation for the work came in the process of localization of a certain web-application, which meant manually translating the texts of all messages and labels used therein. The set of texts that needed translation possessed two notable features. Firstly, it was very restricted – the vocabulary employed was rather narrow and domain-specific. Secondly, the texts were short and contained a certain amount of obvious structure: after having translated the messages "`user added`", "`user deleted`" and "`client added`", the translation for "`client deleted`" could have been derived purely by analogy, without the need for deep lexical, syntactic or semantic analysis. It is clear, that the set of labels and messages is similarly restricted and redundant in most software packages. A method that could exploit the structure and provide at least some aid in the localization process would therefore be of value. The example above immediately suggests an EBMT-like approach, and most probably one of the solutions among, for example, [Nagao, 1984; Kaji et al., 1992; Veale and Way, 1997; Brown, 1996, 2001] would do a good job. However, the cited solutions seem to be, in a sense "too complicated for such a simple case", at least due to the fact that all of these methods rely, in one way or another, on the quality of word-wise alignments of the sentences in the corpus. Therefore, largely out of purely academic interest, an attempt was made to devise a "conceptually simpler" translation model, presented in the following section.

3 The Method

We shall base our translation method on a core observation similar to that employed in all EBMT methods. We refer to it as the *context equivalence principle* and state it informally as follows:

Context equivalence principle *Let AXB and CXD be two phrases in language $\mathcal{L}_1$, and let $\alpha\chi\beta$ and $\zeta\chi\delta$ be their corresponding translations in language $\mathcal{L}_2$. Then, if we find out that the translation of AYB is $\alpha\gamma\beta$, we shall tend to conclude that the translation of CYD is $\zeta\gamma\delta$.*

The logic behind the principle is simple. We assume that the surrounding *context* of the word or a phrase determines the way it should be translated. If we know that X was translated in the same way both in AXB and CXD, we might conclude that it also holds for Y. Note, however, that we do not specify how exactly should the phrase be split into parts A, X and B, and we do not require any "real" semantic relationship between A and α, X and χ, B and β. In this sense, we are completely abstracted from "real languages". We avoid the need for

word-level alignment, and the method can be regarded as a rather general machine learning technique, applicable on arbitrary structures, satisfying the context equivalence principle. In the following we shall define the approach more formally.

We shall represent each sentence in the language as a sequence of *blocks*. Depending on the chosen method of lexical analysis, blocks can be letters, words or some abstract tokens (for example, parsing each word into a tandem of a "stem" block followed by a "form" block can be a good idea). In the experiment we performed, the blocks corresponded to words and punctuation. The set of all blocks used both in the source and the target languages will be denoted by Σ, and the set of all finite sequences of blocks – by Σ^*.

Definition 1 (Language) *A* language $\mathcal{L}$ *is a subset of* Σ^*.

Definition 2 (Translation function) *Let* $\mathcal{L}_1$, $\mathcal{L}_2$ *be two languages. We refer to the map* $t : \mathcal{L}_1 \to \mathcal{L}_2$ *as the* translation function *from the* source language $\mathcal{L}_1$ *to the* target language $\mathcal{L}_2$.

Note that by introducing the definition above we explicitly require each phrase in language $\mathcal{L}_1$ to have a single translation corresponding to it in language $\mathcal{L}_2$. One might argue that this does not always reflect reality, as some phrases might have more than one, and some – no suitable translation. In our case, however, such issues are not of much concern.

Definition 3 (Context, Context translation function) *Let* $\mathcal{L}_1$, $\mathcal{L}_2$ *be two languages and let* $t : \mathcal{L}_1 \to \mathcal{L}_2$ *be a translation function. Let* $A, B \in \mathcal{L}_1$, $\alpha, \beta \in \mathcal{L}_2$. *We call the tuple* $c = (A, B|\alpha, \beta)$ *a* context *of* t, *if, for any* X, *such that* $AXB \in \mathcal{L}_1$ *there exists some* $\chi = t_c(X)$, *such that* $t(AXB) = \alpha\chi\beta$. *The corresponding function* $t_c : \mathcal{L}_{c,1} \to \mathcal{L}_{c,2}$, *(*$\mathcal{L}_{c,1}, \mathcal{L}_{c,2} \subset \Sigma^*$*) is referred to as the* context translation function *of* c.

Let ε denote the empty sequence of blocks. Then it's easy to see, that $c = (\varepsilon, \varepsilon|\varepsilon, \varepsilon)$ is indeed a context of any translation function t, with t_c equal to t. We refer to it as the *trivial context*. It is not necessary for a translation function to have any nontrivial contexts. However, the success of the method relies on the expectation that the translation task to be solved does possess a number of these. It is a plausible expectation, especially in a situation where the freedom of language use is restricted. For example, suppose the language of interest (such as the set of messages in a program) contains a number of phrases of the form "`user ... succesfully`" (e.g. "`user added succesfully`", "`user deleted succesfully`", etc), each of which has the corresponding Estonian translation of the form "`kasutaja edukalt ...`" (e.g. "`kasutaja edukalt lisatud`", "`kasutaja edukalt kustutatud`", ...). Then, c_1 = ("`user`","`succesfully`" | "`kasutaja edukalt`", ε) is necessarily a context of the regarded translation function, as well as, c_2 = ("`user`","`succesfully`" | "`kasutaja`", ε) and c_3 = ("`user`","`succesfully`" $|\, \varepsilon, \varepsilon$). The context translation function t_{c_1} of c_1 maps "`added`" to "`lisatud`", whereas the context translation function of c_2 maps "`added`" to "`edukalt lisatud`". So, once again, note that the whole construction is purely structural and is not strictly required to make sense linguistically. As a final example, consider the identity function t, "translating" from English to English. Then ("`I`", "`home tomorrow`" | "", "`tomorrow`") is a valid context, with the context translation function mapping "`go`" to "`I go home`", etc.

Definition 4 (Context equivalence) *We say that contexts* c_1 *and* c_2 *are* equivalent *if their context translation functions are equal,* $t_{c_1} = t_{c_2}$.

The definitions above formalize the notions, required to exploit the context equivalence principle. They suggest, that in order to translate the phrase AXB, we can find some context $c = (A, B|\alpha, \beta)$ and obtain a correct translation as $\alpha t_{c'}(X)\beta$, where c' is any context equivalent to c, not necessarily c itself. The only thing that remains to be done in order to apply the idea in practice is to perform this operation given only partial information about the translation function in the form of a *training set* of pre-translated sentences, to which we refer as the *empirical translation function*.

Definition 5 (Empirical translation function) *Let t be a translation function between languages $\mathcal{L}_1$ and $\mathcal{L}_2$, and let L be a finite subset of $\mathcal{L}_1$. Let $t_L : L \to \mathcal{L}_2$ be the* restriction of t onto L *(i.e. $t_L(X) = t(X)$ for all $X \in L$). We then refer to t_L as the* empirical translation function *for t.*

The empirical translation function encapsulates all the assumption-free information about the translation function present in the training set, and we can already use it as a CAT tool (this idea is often referred to as *translation memory* (TM) [Schäler, 2001] and is indeed implemented in the majority of CAT tools). However, as noted above, the structure inherent to languages that satisfy the context equivalence principle could allow to extract more than that, if we only knew the contexts of t and their pairwise equivalences. Of course, t_L doesn't contain enough information to infer required knowledge with absolute certainty. However, we can still infer it with *partial* certainty.

Definition 6 (Empirical context, Empirical context translation function) *Let t_L be an empirical translation function and let the* confidence parameter *n be a positive integer. We say that a tuple $c = (A, B|\alpha, \beta)$ is an* empirical context of t, according to t_L with confidence n, *if:*

1. *c is a context of t_L,*
2. *There exist at least n different X, such that $AXB \in L$.*

The context translation function of c (with respect to t_L) is referred to as the empirical context translation function of c.

For example, if the training set contains translations of "`user added succesfully`", "`user deleted successfully`" and "`user logged in successfully`", and *all* of the translations of "`user ... successfully`" are of the form "`kasutaja edukalt ...`", then we might conclude, that $c =$ ("`user`", "`succesfully`"| "`kasutaja edukalt`", ε) is a context of t. As we are not really sure (we haven't seen the whole t), we refer to it as an *empirical context of t with confidence 3*. We can deduce what mappings the corresponding context translation function would have for "`added`", "`deleted`" and "`logged in`", and we encapsulate this information in the *empirical context translation function* of c.

In a similar spirit we define empirical context equivalence.

Definition 7 (Empirical context equivalence) *Let t_L be an empirical translation function for t, and let c_1, c_2 be two empirical contexts with some confidence n. Let $t_{c,1}$ and $t_{c,2}$ be the empirical context translation functions of c_1 and c_2. Let C be the intersection of the domains of these functions (i.e. all block sequences that can be translated by both functions). We say that c_1 and c_2 are* empirically equivalent with confidence $m > 0$, *if:*

1. *$t_{c,1}(X) = t_{c,2}(X)$ for all $X \in C$,*
2. *$|C| \geq m$.*

By continuing the example above, if the training set would also contain translations of "`client added`", "`client deleted`" and "`client is offline`", all of the form "`klient ...`", and the context translation of "`added`" and "`deleted`" would equal that of the above shown empirical context ("`user`", "`succesfully`"| "`kasutaja edukalt`", ε), we could conclude that:

1. ("`client`", ε | "`klient`", ε) is an empirical context with confidence 3.
2. ("`client`", ε | "`klient`", ε) is empirically equivalent to ("`user`", "`succesfully`"| "`kasutaja edukalt`", ε) with confidence 2.

By appealing to the context equivalence principle we now could translate phrases "`client logged in`" and "`user is offline successfully`", although none of these were present in the original training set. Of course, the meaninglessness of the latter phrase can not be detected by the algorithm, as it is impossible to deduce the required semantic information from the scarce training set without external knowledge.

It is worth noting, that empirical equivalence is not transitive and is thus not a "true" equivalence relation. However, it does not prevent its use in practice.

By now we have constructively defined the way to detect empirical contexts, compute empirical equivalence, and use the context equivalence principle for translation. We summarize it in the following algorithm:

Algorithm 1 (Context equivalence-based MT) *Given a training set t_L a phrase to be translated P, and the confidence parameters n and m, do the following:*

1. *Compute all empirical contexts of t_L with confidence n.*
2. *Compute the empirical equivalence with confidence m between each pair of detected empirical contexts.*
3. *Find all empirical contexts, whose left part matches the phrase. For example, if the phrase can be represented as $P = AXB$, then all contexts of the form $(A, B|\cdot, \cdot)$ fit.*
4. *Suppose $P = AXB$ and $c = (A, B|\alpha, \beta)$ is a matching empirical context. If the context's translation function t_c can translate X, append $\alpha t_c(X)\beta$ to the list of suggested translations of the phrase P. Also for each empirically equivalent context c', whose translation function $t_{c'}$ can translate X, append $\alpha t_{c'}(X)\beta$ to the list of translations.*
5. *If the resulting list of translations contains a single element, report it as the correct translation, otherwise declare the phrase to be untranslated.*

Note that it is possible at step 5 of the algorithm for the list to contain several translations. This happens when several contexts, empirically equivalent to c are not empirically equivalent among themselves, and in this case some of the translations in the list are most probably wrong. Instead of attempting to figure out which of the translations is the correct one, our algorithm behaves conservatively and produces no answer.

The question of efficient implementation of the algorithm is a separate research problem and is not in the scope of this paper.

4 Results

A more-or-less straightforward version of the algorithm was implemented. As we are not dealing with the issues of efficient implementation of the strategy here, we omit the discussion of the questions of performance and optimization, and only note that a reasonably straightforward python implementation, when run on a dataset of about 1500 training instances on a contemporary laptop computer spends about 4 minutes on precomputation. It is enough for research purposes and batch translation, but definitely needs further work if a production-state implementation were wished for.

We tested the algorithm on two datasets in a number of settings. The first dataset consisted of 1623 messages from a web application in Estonian and Russian languages in the form of a gettext .PO file [Gettext]. Each message was split into words and punctuation. We performed a leave-one-out cross-validation (LOOCV) on this dataset, that is, for each instance, we used 1622 remaining instances to deduce the correct translation for the chosen message, and compared it to the original translation. We used 2 as the confidence parameter for both empirical context detection and empirical context equivalence. The results were as follows:

1158	instances had no suggested translation
16	instances had multiple suggested translations
433	instances had translations exactly matching the correct ones
13	instances had correct translations, that didn't exactly match the ones in the training set. In fact, about 7 of these were better than the ones in the training set.
3	instances were translated incorrectly. The mistake was related to the gender-specific declination of an adjective, which is not present in Estonian, yet is present in Russian.

To summarize, about 27.5% of instances were translated *precisely* as a human translator would, and only 3 translations would require minor correction – an error rate comparable to a human translator. The number of untranslated messages is high, yet it corresponds rather closely to the initial intuition obtained when manually translating the messages. Intuitively, it seemed that roughly half of all the messages definitely needed to be translated manually in order to set the domain-specific terminology and translator-specific style. The remaining half seemed to contain sufficient amount of redundancy in order to be translatable automatically. The fact that more than 27% of messages could be translated with the help of a straightforward substitution-based principle with nearly no mistakes seems thus to be very satisfactory. Moreover, if we relax the confidence parameter for empirical context equivalence from 2 to 1, the precision rate goes up to 36.8% and the number of erroneous translations to 10 (for complete statistics consult the Appendix).

We have repeated the above experiment on another dataset, that was obtained by merging together the English-Estonian translations of the messages used in the *KDE Games* Linux package [KDEGames]. The resulting dataset contained 2620 messages. Of these, 369 (14.1%) could be appropriately translated with 7 errors when the context equivalence confidence 2 was used. With confidence parameter 1 the precision rate was 20.7% and the number of errors – 13. One reason why the results here are significantly worse than for the web application dataset is lower coherence of the phrases. The KDE dataset was a collection of translations from several different games translated by different people, and the ways of expression differed slightly among the games.

The results of the leave-one-out cross-validation experiments do indicate the percentage of "easily translatable" messages within the dataset, but this provides only an overly optimistic estimate of the actual utility of the method. Indeed, consider the dataset containing the trans-

lations of the four messages: "`user added`", "`user deleted`", "`client added`" and "`client deleted`". It may very well be possible, that the translation of any of the four can be derived from the 3 other translations, hence the LOOCV result would be 100%, yet this number does not indicate the usefulness of the method in practice, which would be better estimated as 25% in this case. In order to assess the "actual performance" of the approach we have conducted a split-set experiment: a random subset of instances was selected for training, and the remaining instances were given to the algorithm to be translated. The precision rates were expectably worse.

For the web-application dataset we have randomly selected 1000 of the 1623 instances for training. Of the remaining 623 instances, 37 (5.9%) were succesfully translated with confidence parameter 2, and 66 (10.6%) – with confidence parameter 1 (details in the Appendix). For the KDE dataset the results were 5.5% and 7.4% correspondingly (2000 instances were used for training and 620 – for testing).

The split-set results do look much less inspiring than the LOOCV ones, yet they still indicate the potential of the method to perform better than the bare "translation memory" approach. A smart choice of the training set as well as the use of external linguistic knowledge might further boost the utility of the approach without significantly compromising the simplicity of the underlying framework, yet the consideration of these issues is, unfortunately, out of the scope of this paper.

5 Discussion

We have introduced a novel example-based machine translation method. The proposed method was motivated by an observation, that the set of labels and messages of a software package possesses a lot of redundancy and structure. This redundancy might allow a very simple approach to somewhat reduce the job of the human translator by producing reliable translations by analogy with the already created ones. Experiments have shown, that it is indeed the case, and the translation of about a quarter of the messages in the considered example can be derived by analogy from the other translations.

The method uses no linguistic prior knowledge, yet it should be noted that in the considered examples, all of the empirical contexts with at least one other equivalent context made sense linguistically. It suggests that the approach can be used as a basis for a method of grammar extraction or vocabulary construction.

The idea of stricty following a single simple heuristic certainly contributed strongly to the method's low mistake count. However, it is conceivable that other simple heuristics exist that might result in better performance. For example, the current approach does not detect analogies where the interchangeable element is not a continuous range of blocks within a phrase (e.g. $AXBXC \rightarrow \alpha\chi\beta\chi\zeta$). It is straightforward to generalize the idea of context equivalence for this case, yet its efficient implementation is less obvious.

Another direction for further research might be related to the problem of incorporating external linguistic knowledge into the described framework. The presented translation method uses 2 notions – context equivalence and context translation functions. Whereas the latter should be specific to the domain and the choices made by the human translator, the former is largely dictated by the grammars of the source and target languages and it should be possible to derive context equivalence by some decently simple syntactic analysis. This might significantly boost the performance of the method without cluttering the framework with ad-hoc heuristics.

Finally, the issues of practical application of the approach are worth consideration. For example, it would be nice to have a computer-assisted software localization system, that could:

- keep a set of messages translated earlier, potentially from other projects.
- present the untranslated texts to the user in the optimal order, so that new automatic translations could be derived and checked as early as possible.
- detect when some of the earlier translated messages from a different project fit badly with the new translations and drop them from the training set.

The above vision, however, presents some significant challenges, that are of interest both from the practical, as well as from the academic points of view. For example, the problem of optimally selecting the training set may well be NP-complete.

To conclude, the developed formal framework, despite its simplicity, seems to provide ample space for further research, and the presented translation method, although quite raw yet, can already be applicable in some specific practical situations.

References

R. Brown. Example-based machine translation in the pangloss system. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 169–174, 1996. URL `http://www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. Transfer-Rule Induction for Example-Based Translation. In *Proceedings of the Workshop on Example-Based Machine Translation*, September 2001. URL `http://www.iai.uni-sb.de/~carl/ebmt-workshop/`.

Gettext. Gettext — wikipedia, the free encyclopedia, 2007. URL `http://en.wikipedia.org/w/index.php?title=Gettext&oldid=120568329`.

Hiroyuki Kaji, Yuuko Kida, and Yasutsugu Morimoto. Learning translation templates from bilingual text. In *Proceedings of the 14th conference on Computational linguistics*, pages 672–678, Morristown, NJ, USA, 1992. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/992133.992174.

KDEGames. Kde games website. URL `http://games.kde.org/`.

Ruslan Mitkov, editor. *Machine translation: general overview*, chapter The Oxford Handbook of Computational Linguistics., pages 501–511. Oxford University Press, 2003.

Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc. ISBN 0-444-86545-4.

Reinhard Schäler. Beyond translation memories. In *Proceedings of the Workshop on Example-Based Machine Translation*, 2001. URL `http://citeseer.ist.psu.edu/691661.html`.

StatMT. Statistical machine translation website. URL `http://www.statmt.org/`.

Davide Turcato and Fred Popowich. What is example-based machine translation? In *Proceedings of the Workshop on Example-Based Machine Translation*, 2001. URL `http://citeseer.ist.psu.edu/685499.html`.

Tony Veale and Andy Way. Gaijin: A bootstrapping, template-driven approach to example-based mt. In *Proceedings of the NeMNLP97*, 1997. URL `http://citeseer.ist.psu.edu/723936.html`.

All URL-s referenced above were still valid on the 15th of May, 2007.

Appendix

Results of the conducted experiments are presented in the table below. The meaning of the columns is the following:

Data The dataset used: either the 1623 Estonian/Russian messages from a web-application or the 2620 messages from KDE Games.

Test The test performed: either LOOCV or split set. In the case of the web application, 1000 instances were used for training, and in the case of KDE games – 2000.

Conf The confidence parameter used to detect empirical context equivalence. Note that the second confidence parameter, the one used to detect empirical contexts, was always set to 2.

Total Total number of tested instances.

Untr Number of instances for which the algorithm did not find any translation.

Mult Number of instances the algorithm could not translate because several translations were detected.

Exact Number of instances, for which the algorithm found the translation exactly matching the true one.

Inexact Number of instances, for which the algorithm found the translation, which did not exactly match the true one, yet was still acceptably correct (in many cases better than the true one).

Wrong Number of instances, for which the algorithm reported a wrong translation.

Prec Precision rate: (Exact+Inexact)/Total.

Data	Test	Conf	Total	Untr	Mult	Exact	Inexact	Wrong	Prec
Webapp	LOOCV	2	1623	1158	16	433	13	3	27.5%
KDE	LOOCV	2	2620	2218	26	361	8	7	14.1%
Webapp	Split	2	623	578	6	34	3	2	5.9%
KDE	Split	2	620	571	15	28	6	0	5.5%
Webapp	LOOCV	1	1623	957	58	583	15	10	36.8%
KDE	LOOCV	1	2620	1970	94	527	16	13	20.7%
Webapp	Split	1	623	540	12	54	12	5	10.6%
KDE	Split	1	620	560	8	32	14	6	7.4%