ISBN 978-83-60810-22-4
ISSN 1896-7094

It's all about the Trees
Towards a Hybrid Syntax-Based MT System

Marcin Junczys-Dowmunt
Adam Mickiewicz University
Faculty of Mathematics and Computer Science
ul. Umultowska 87, 61-614 Poznań, Poland
Email: junczys@amu.edu.pl

***Abstract*—The aim of this paper is to describe the first steps of research towards a hybrid MT system that combines the strengths of rule-based syntactic transfer with recently developed syntax-based statistical translation methods within a unified framework. The similarities of both paradigms concerning the processing of syntactically parsed input trees serve as a basis for this reseach. We focus on the statistical part of the future system and present a syntax-based statistical machine translation system—BONSAI—for Polish-to-French translation. Although BONSAI is still under develepmont, it reaches a translation quality on par with that of a modern phrase-based system. We provide the theoretical background as well as some implementation details and preliminary evaluation results for BONSAI. At the end of this paper we shortly discuss the benefits of a combined approach.**

I. INTRODUCTION

ACADEMIC research in machine translation of the last decade has been focused on statistical machine translation (SMT), especially on the phrase-based variants of the paradigm, while the classical rule-based approach is marginalized.

In the sector of translation software available for home users, however, the rule-based approach prevails. A quick look at [1] reveals only one commercial product for desktop computers—PLAINTRANSLATE[1]—that uses statistical translation techniques. LANGUAGE WEAVER[2] offers a commercial MT software package that is entirely based on SMT, but here a client-server architecture is required. The translation is conducted by a central unit and the software installed on the client computer is not involved in the translation process. The reasons for such a centralized approach become obvious when we read that the installation of only one additional language pair requires about 30 GB of free disk space.

While SMT seems to exploit resources which even today a typical home user will not give up voluntarily to a single piece of software, the requirements of rule-based MT systems are rather modest, both in terms of memory usage and processing time. Recent tests [2] showed that when it comes to human evaluation, rule-based system can achieve results not worse or better than SMT systems, even if the automatically calculated BLEU scores seem to suggest different conlusions.

Manually designed translation rules scale up well to general translation tasks and are fairly domain-independent. At the same time, it is quite easy to fine-tune the translation of particular syntactic or lexical phenomena although this may become problematic if many such phenomena are to be considered. Modern syntax-based statistical methods, on the other hand, provide at minimum cost millions of high quality domain-dependent translation rules that are triggered by signal words and can therefore model context-specific syntactic transformations. Nevertheless SMT suffers the consequences of all corpus-based methods: the handling of unseen structures, words or meanings. This is especially true if resources are scarce, as is the case with parallel corpora involving Polish, our main language of interest. Here the manually built translation lexicons of a rule-based system—usually of a high quality and coverage—might prove helpful.

In this paper we present the first steps of research towards a hybrid MT system that combines the strengths of rule-based syntactic transfer with recently developed syntax-based statistical translation methods. The similarities of both paradigms concerning the processing of syntactically parsed input trees serve as a basis for this research. The rule-based system we plan to adapt in the future is an existing commercial MT application—TRANSLATICA—that will be introduced in the next section. We focus on the statistical part of the future system and present a syntax-based statistical machine translation system—BONSAI—for Polish-to-French translation. BONSAI works currently as a stand-alone application and although it is still under develepmont, it reaches a translation quality on par with that of a modern phrase-based system.

An overview of syntax-based methods for SMT is provided in section III. Sections IV to VII give the theoretical background, implementation details and evaluation results for BONSAI. Source language analysis that is a common necessary step for both, rule-based MT and syntax-based SMT, is addressed in section V. In the same section we illustrate the process of automatic extraction of translation rules.

Section VIII highlights the similarities between rule-based syntactic transfer and syntax-based statistical machine translation of the type decribed in this paper. We outline briefly how both types of systems can interact with each other due to their similarities and benefit from each other due to their different approaches.

The author of this work has received a scholarship from the Adam Mickiewicz University Foundation for the year 2009.

[1]http://www.apptek.com/index.php/plaintranslate
[2]http://www.languageweaver.com

II. The rule-based reference system

Translatica [3] is a rule-based commercial MT system continuously developed by Poleng Sp. z o o.[3] since 1995. Since Translatica is a product tailored to the needs of the Polish market, it allows translations from and to Polish for English, Russian and German. According to the Vauquois Triangle, Translatica has to be classified as a MT system based on syntactic transfer with elements of direct and semantic transfer. A large and richly annotated translation lexicon contains a great number of idiomatic phrases and extensive semantic information for all its entries, e.g. an ontology reminiscent of a bilingual wordnet or information concerning the translation of verb frames. Until now, the transfer rules as well as the translation lexicons for all supported language pairs have been created manually—those for the Polish-German language pair by the author of this paper.

The paradigm of syntactic transfer requires the input to be syntactically parsed. The parser [4] built into Translatica relies on a manually created context-free grammar. A mechanism similar to feature unification is supported but restricted to one level of argument depth. Thus the expressive power of the grammar is not increased, but it is easier to design rules by hand, e.g. case agreement can be set by attributes instead of using complex symbol names. The parser has been extended with the possibility to perform basic semantic role labelling. Preprocessing performed before parsing includes named entity recognition, identification of multi-word units, and idiomatic phrases.

Syntactic transfer in Translatica relies on parse tree transformations which are encoded in a specialized programming language. Default operations specified for each syntactic category are executed in a recursive depth-first post-order fashion in three different passes. During the first pass various normalization steps are performed, e.g. renaming of source language syntactic categories to target language categories, the deletion of source language negation particles, or a head-wise up-propagation of morphological features. Re-orderings of child nodes, insertions of new target language nodes and the down-propagation of target language features are performed in the second pass. The third pass involves the generation of correct morphological forms of the target language tokens considering, for instance, the agreement of number, gender, or case between dependent target words, etc.

From the depth-first postorder processing scheme follows that in the general[4] case the children of a node are processed independently from each other and before their parent node. Actions that concern two or more subtrees rooted in sibling nodes[5] are therefore carried out by operations attached to their parent node.

[3]http://www.poleng.pl

[4]Child nodes can refer to their ancestors by special instructions, but this is rarely used and must be invoked explicitly.

[5]Examples for such actions are re-orderings, combined down-propagation of morphological features of a verb and its complements following from verbal frames, etc.

III. Syntax-based SMT

By **syntax-based** SMT we mean methods based on statistical translation models that incorporate hierarchical syntactic representations—i.e. syntactic parse trees of any form—of the source and/or the target language. Factored phrase-based models as introduced in [5] that use syntactic or morphological information as factors do not match this requirement, although they are sometimes referred to as syntax-based models.

One of the first syntax-based SMT approaches in this sense is [6] where parsed source sentences form the input for an English-to-Japanese translation process. The translation model consists of probabilistic permutation, deletion, insertion, and translation tables defined for tree nodes and their children. These tables are obtained from a word aligned and source language parsed parallel corpus.

The concept of phrase-based models is extended to a hierarchical phrase-based model by [7]. A Chinese-English **synchronous context-free grammar** (SCFG) with only one (recursively used) non-terminal symbol X is automatically created extending the approach for flat phrase-based models from [8]. This SCFG is used for source language parsing which results in a synchronous parse tree (**SMT by parsing**). The target sentence can then be read off the target language part of the parse tree's leaves. The difference between this method and all other approaches presented in this section is that the corpus used for training has not been syntactically parsed. The SCFG is computed from the raw alignment data alone.

The idea of translation by parsing is further enhanced by [9] who train a French-English SCFG on a word aligned parallel corpus the target language part of which has been syntactically parsed. A chart parser is used to build target language parse trees by parsing the source language with the SCFG. Compared to the previous method, the SCFG employs all non-terminal symbols of the parser used for annotation as well as incomplete nonterminals that correspond to the non-syntactic phrases of a phrase-based system.

Instead for parsing, SCFGs and similar concepts can be used for the direct translation of syntactic trees created independently from the SCFG. The idea dates back to [10] where so called tree transducers are employed for the translation of programming language parse trees to machine code. A theoretical framework for tree transducers is presented in [11]. The model from [6] is later formalized as a tree-to-tree transducer in [12]. In this approach, tree transformation rules are restricted to the first level of the matched subtree's structure, i.e. only the direct children of a subtree root can be matched and changed by the application of a single rule. [12] also introduce **extended-LHS** tree transducers which are related rather to synchronous tree-substitution grammars (STSG) [13] than to SCFG. In an extended-LHS tree transducer, single production rules can perform transformations of trees on multiple levels of the tree structure without being restricted to the direct children.

A variant of extended-LHS tree transducers—extended-LHS tree-to-string transducers (**xRs**)—are described by [14] in the context of an English-Chinese MT system. In the following sections we will present a similar approach for a syntax-based Polish-to-French SMT system. Compared to [14], we use a flatter representation of translation rules by modifying the concept of **xRs**, as will be illustrated below. As in [6] and [12], **xRs** require the input to be a syntactic tree or a string of such trees. The final output consists of plain target language strings.

We implemented a syntax-based SMT system similar in principle to the models presented in [14] and [15] and based on the theory from [11]. The working name of the system used in this paper is BONSAI. The system is developed under the assumption that it will be combined with a commercial rule-based MT system. In this paper we concentrate on Polish-to-French translations, for the Polish-French language pair is the next to be integrated into TRANSLATICA. The next section describes the formal aspects of the system in more detail.

IV. DESCRIPTION OF THE TRANSLATION MODEL

Throughout this work we will traditionally denote the source language sentence by $\mathbf{f}$ and the target language sentence by $\mathbf{e}$. Strings consist of source language parse trees, target language symbols, or combinations thereof.

A. Tree-to-string transducers

By T_Σ we denote the set of ordered, finite, and rooted trees over an alphabet Σ. Formally our translation algorithm is a tree-to-string transducer. For an exhaustive theoretical description of tree transducer see [11]. Following [11] we define a *weighted extended-lhs root-to-frontier tree-to-string transducer* M as a 5-tuple $(\Sigma, \Delta, Q, q_0, R)$, where

- Σ is the input alphabet,
- Δ is the output alphabet,
- Q is a finite set of states,
- $q_0 \in Q$ is a distinguished start state,
- R is a finite set of productions of the form $(q, \delta) \rightarrow^w rhs$.

Here δ is a pattern for the LHS tree that checks whether a given subtree can be matched by the rule. The tree's root is required to have been marked by the state $q \in Q$. The leaves of a LHS tree are labelled with terminal symbols from Σ or a variable from the set of variables $\mathcal{X} = \{x_1, x_2, x_3, \ldots\}$. The right-hand side $rhs \in (\Delta \cup (Q \times \mathcal{X}))^*$ is a string of target language terminals and those variables that appeared and were matched in the pattern. Each rule is associated with a weight $w \in \mathbb{R}^+$.

The tree transducer operates on strings consisting of source language trees and target language symbols from the set of strings $(\Delta \cup (Q \times T_\Sigma))^*$. A Polish source sentence corresponds to a string that consists only of source language trees paired with a state from the transducer. A French target sentence is a string of symbols from the target alphabet. Intermediate results will contain source language trees as well as target language symbols.

A derivation is a triple (a, b, h), where a and b are strings of the type described above and $h \in (\mathbb{N} \times R)^*$ is the derivation history, a sequence of string indexes and productions rules from the tree transducer that were applied for the transduction of a to b. We are interested in the set of left-most derivations $LD(M)$, a subset of the set of derivations $D(M)$, where productions are always applied to the left-most source language tree in a string. Again, for the full formal description of $D(M)$, $LD(M)$ and the corresponding derivation relations see [11].

The productions of an **xRs** can match and transform trees of arbitrary height and complexity, but for the matching function δ to match a particular subtree the complete structure of the subtree must be specified (compare [14]). We modify the concept of **xRs** by simplifying δ, the matching function. Instead of matching the whole subtree structure it is sufficient to match the root of the subtree and the sequence of descendants of the specified root that are actually modified by the rule. This sequence of descendants has to comply with the following two criteria:

- no two nodes are descendants of each other;
- the concatenation of the yields of the subtrees rooted in these nodes is equal to the yield of the whole subtree.

We call any such a sequence a **fringe**. Thus the internal tree structure is ignored and productions can be represented without redundant information. This seems to be a reasonable modification of the **xRs** concept especially for tree-to-string transducers where the tree structure is increasingly flattened with each step. If the matching part of the first rule applied to the example tree from Fig. 1 were fully specified, it would have to take the following form:

S(ADV(obecnie), VP(x_0:V-PP, V(są)), x_1:NP).

Instead we can simply write

S(obecnie, x_0:V-PP, są, x_1:NP).

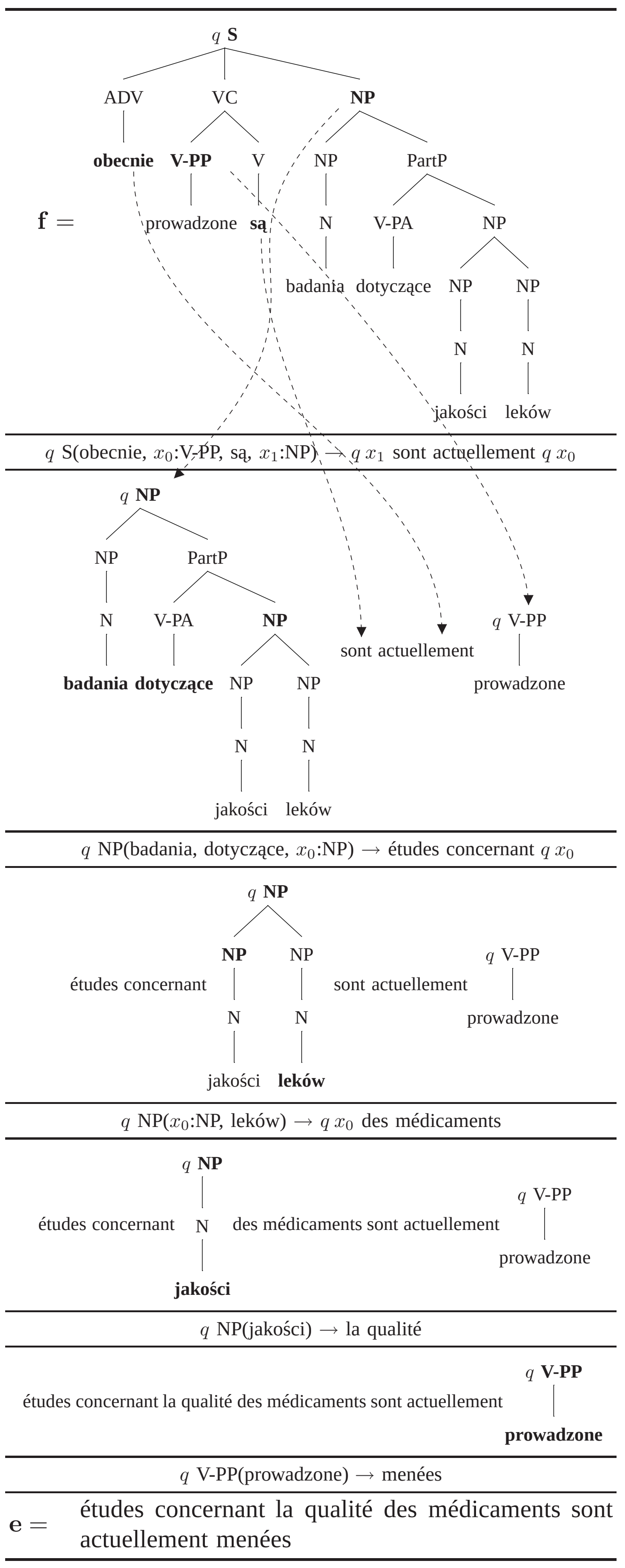
This notation is especially useful for purely lexical rules that do not contain non-terminal symbols. The fringe is then equal to the yield of the matching subtree. The tree pattern

PartP(V-PA(dotyczące), NP(NP(R(jakości), NP(R(leków)))))

can be represented only by the labels of its leaves as

PartP(dotyczące, jakości, leków).

In theory, such a rule can match several non-isomorphic trees, but in practice, it is not very likely unless the context-free grammar used for parsing is very ambiguous. Even if it does happen it should not do much harm to the translation result. For rules that consist only of terminal symbols, we can safely assume that the internal tree stucture is irrelevant. In the case of mixed rule the terminal symbols should provide enough context to prevent incorrect translations. An **xRs** modified in such a way is of course not suited for the transduction of parse trees of formal languages where ambiguity should be avoided. For natural language processing, however, where allowing for a little vagueness can even be helpful, this slightly more compact reprensentation can save memory and matching time if the **xRs** consists of millions of rules.



f =
q S
ADV VC NP
obecnie V-PP V NP PartP
prowadzone są N V-PA NP
badania dotyczące NP NP
N N
jakości leków
q S(obecnie, x_0:V-PP, są, x_1:NP) → $q\ x_1$ sont actuellement $q\ x_0$
q NP
sont actuellement
q V-PP
prowadzone
q NP(badania, dotyczące, x_0:NP) → études concernant $q\ x_0$
études concernant
q NP(x0:NP, leków) → $q\ x_0$ des médicaments
des médicaments sont actuellement
q NP(jakości) → la qualité
études concernant la qualité des médicaments sont actuellement
q V-PP(prowadzone) → menées
e = études concernant la qualité des médicaments sont actuellement menées

Fig. 1. A sample derivation with heavy reordering of terminal and non-terminal symbols. The bold labels mark the symbols on the fringe that are matched by the rule given below each intermediate derivation.

Fig. 1 illustrates a sample derivation of the following Polish sentence:

Glosses:

obecnie	*prowadzone*	*są*	*badania*	*dotyczące*	*jakości*	*leków*
actuellement	menées	sont	études	concernant	la qualité	des médicaments

Correct: études concernant la qualité des médicaments sont actuellement menées

English: studies concerning the quality of medicaments are currently being conducted

Polish is a free-word-order language and the above glosses imply that a lot of re-ordering has to be performed to bring the fragments of the French translation into the correct order. In the example translation this is achieved by the first rule. It works on the sentence level and shifts around the main constituents of the sentence (movements marked by dashed arrows). Similar re-orderings are much harder to achieve in a phrase-based statistical machine translation system.

The costs of a derivation (a, b, h) where h is the derivation history of length n is calculated as follows:

$$w_M((a, b, h)) \equiv \prod_{i=1}^{n} w_i. \tag{1}$$

Different derivations can yield the same transduction results for a pair of strings a, b. Therefore, the global derivation cost for a, b is calculated as

$$W_M((a, b)) \equiv \sum_{(a,b,h)\in LD(M)} w_M((a, b, h)). \tag{2}$$

Based on this, we can formulate a translation model based on a tree-to-string transducer M. The best translation of a (parsed) source sentence $\mathbf{f}$ is the target sentence $\mathbf{e}$ that can be derived from $\mathbf{f}$ by M (in a left-most fashion) and that maximizes the global derivation cost W_M:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} W_M((\mathbf{f}, \mathbf{e})) \tag{3}$$

Obviously there is no target language model integration yet. The only information that guides the translation process in such a model are the syntactic annotations of the source sentence and the costs of the production rules.

B. Rule features

The production rule cost w is split into several feature costs. For each translation rule we actually gather the following information:

- the rule probability conditioned on the source language part of a rule;
- the rule probability conditioned on the target language part of a rule;
- the source conditioned word translation probability which estimates whether the source and target words of a rule are good translations of each other (see [8] for a detailed description);

- the target conditioned word translation probability;
- the non-terminal symbol penalty $e^{-\frac{k^2}{n}}$ where k is the number of non-terminal symbols in a rule. That way the use of lexical rules is favoured over compositional translations;
- the length difference penalty $e^{-|\alpha n - m|}$ where n and m are the lengths of the source and target language parts of a rule respectively. α is the average ratio of target sentence and source sentence lenghts estimated from the training corpus. For Polish-to-French translation we have $\alpha = 1.22$, i.e. French sentences have in average 1.22 times more words than Polish sentences.

The total cost w of a production rule $q\ \delta\ \rightarrow^{w}\ rhs$ is then calculated as $w = \prod_{j=1}^{k} c_j^{\lambda_j}$, where c_j are the above mentioned rule costs weighted by a parameter λ_j. These parameters are the main parameters of the translation model.

C. Approximation of the best translation

With millions of translation rules it is not feasible to calculate the global derivation cost for a source sentence $\mathbf{f}$ and a target sentence $\mathbf{e}$. Therefore we decided to approximate the globally best translation by the single best derivation:

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} W_M((\mathbf{f},\mathbf{e})) \\ &\approx \arg\max_{\mathbf{e}} \Big(\max_{(\mathbf{f},\mathbf{e},h)\in LD(M)} w_M((\mathbf{f},\mathbf{e},h)) \Big).\end{aligned} \tag{4}$$

Instead of searching for a best translation $\hat{\mathbf{e}}$, we search for the best pair $(\hat{\mathbf{e}}, \hat{h})$. Now formulating the translation model as a log-linear model is simple: for a given parsed source sentence $\mathbf{f}$ we find the best translation $\hat{\mathbf{e}}$ based on the best derivation history $\hat{h}$ by

$$\begin{aligned}(\hat{\mathbf{e}}, \hat{h}) &= \arg\max_{(\mathbf{e},h)} w_M((\mathbf{f},\mathbf{e},h)) \\ &= \arg\max_{(\mathbf{e},h)} \prod_{i=1}^{n} w_i \\ &= \arg\max_{(\mathbf{e},h)} \prod_{i=1}^{n}\prod_{j=1}^{k} c_{j_i}^{\lambda_j} \\ &= \arg\max_{(\mathbf{e},h)} \sum_{j=1}^{k} \lambda_j \underbrace{\sum_{i=1}^{n} \log c_{j_i}}_{\log p_j(\mathbf{e},\mathbf{f},h)}\end{aligned} \tag{5}$$

where $h = ((p_1, r_1), \ldots, (p_n, r_n))$, $r_i = (q_i, \delta_i, rhs_i, w_i)$, and $(\mathbf{f}, \mathbf{e}, h) \in LD(M)$. Here p_j can be interpreted as the cost of a single feature for the complete derivation h and λ_j becomes an overt parameter of the model that is not hidden in the weights of particular productions.

D. Complete model

Within a log-linear model we can easily enrich the model with additional features. For the moment, only a basic language model p_{lm} over target language words is added to the model. Finally, for a parsed source language sentence $\mathbf{f}$ we have:

$$(\hat{\mathbf{e}}, \hat{h}) = \arg\max_{(\mathbf{e},h)} \Big(\sum_{j=1}^{k} \lambda_j \sum_{i=1}^{n} \log c_{j_i} + \lambda_{k+1} \log p_{\mathrm{lm}}(\mathbf{e}) \Big) \tag{6}$$

where $h = ((p_1, r_1), \ldots, (p_n, r_n))$, $r_i = (q_i, \delta_i, rhs_i, w_i)$, $w_i = \prod_{j=1}^{k} c_{j_i}^{\lambda_j}$, and $(\mathbf{f}, \mathbf{e}, h) \in LD(M)$.

Our translator is thus an extended-LHS tree-to-string root-to-frontier transducer with a single state q. The semantics of this state can be paraphrased as *apply a production rule to this subtree*. The input alphabet Σ consists of Polish words found in the source language half of a parallel training corpus and of non-terminal symbols from the grammar used for source language parsing. The output alphabet Δ consists of French words from the training corpus only, i.e. there are no non-terminal symbols in Δ. We still need to describe the set of productions R, the construction of which will be discussed shortly in the next section.

V. PARSING AND RULE EXTRACTION

For the extraction of productions, the source language part of the entire training corpus needs to be parsed by the same parser that will be used during the translation. If a corpus consists of millions of sentences, this is sure to be very time consuming. Currently, the parser used in TRANSLATICA is strongly entangled with the transfer process and cannot be easily used as an independent module for the annotation of corpus data or as the parser of BONSAI. Work on the extraction of the parser from the surrounding code is already in progress.

Since TRANSLATICA's deep parser is at the moment unavailable, we decided to use the shallow parser SPEJD [16] for our experiments. SPEJD's rules form a cascade of regular expressions that upon match combine tokens or previously built groups into larger groups. These groups are annotated with syntactic categories and morphological features of their specified heads. Apart from building hierarchical structures, SPEJD allows for the simple unification of morphological features which can also be used as a disambiguation mechanism for morphologically ambiguous words. This is a very helpful feature if a strongly inflectional language like Polish is analyzed.

The grammar used for parsing has been written from scratch and consists of no more than 274 rules, of which only 68 are used for the construction of syntactic parse trees. The remaining rules are descriptions of multi-word expressions that are to be treated as single tokens, an example is the expression *choćby dlatego aby* (eng. *just so that*) which should be marked as a single conjunction which joins sentences but not phrases. It is however clear that the use of a shallow parser with such a small grammar comes at some costs: the height and complexity of the parse tree is restricted by the number of rules in the grammar since no true recursion is possible and problems like PP-attachment or attachment issues in general are virtually unsolvable within the formalism. We therefore try to keep the resulting trees very flat with chunks being attached at sentence level. Nevertheless, complex parses are possible and often impressively correct like the example parse tree from Fig. 1 that has been produced by SPEJD. It can further be assumed that systematic and consistent parse errors do not necessarily decrease translation quality since the translation rules can in turn learn the correct translations systematically. One huge advantage of a shallow parser is speed: millions of sentences can be analyzed within a few hours which simplifies the development phase.

Once the source language part of a parallel corpus has been analyzed by SPEJD, we extract productions in a similar way as [9], but restrict them to complete syntactic phrases. The parallel corpus is word-aligned by training GIZA++ in both directions. Prior to word alignment we lemmatize both sides of the corpus. Experience shows that lemmatization can significantly improve alignment quality if at least one language is strongly inflectional, as is the case with Polish. The resulting alignments are combined with the refined symmetrization method described in [17] and lemmatization is undone. For each aligned sentence the source parse tree is connected with the alignment graph. Next we find all phrase pairs that are consistent with the word alignment [18] and a subtree of the parse tree. For all subtrees in compliance with these criteria the set of fringes is generated and the alignment data is used to create a production for each fringe.

The number of fringes grows exponentially with the size of the tree, hence, we apply the following restrictions:

- the fringe consists of no more than 7 symbols of which at most 4 are non-terminals;
- the distance of a non-terminal in the fringe from the root is at most 2.

The described procedure has been applied to the Polish-French part of the JRC-ACQUIS parallel corpus [19] with about 1.2 M sentences. In spite of the mentioned restrictions on average 130 rules per sentence are generated. This results in a set of 50 millions unique rules occupying 5.5 GB of disk space. Compressing them to a minimal finite state automaton reduces the size to 1.3 GB.

One problem with our approach is the possibly high redundancy of rules. An alternative method is presented by [20] who try to extract only a minimal set of translation rules that sufficiently explain the translation data in a parallel sentence pair. It needs to be examined how the different approaches affect the size of the rule set and the translation quality.

VI. SEARCHING FOR THE BEST DERIVATION

For a given string of source trees $\mathbf{f} \in (Q \times T_\Sigma)^*$ the subset of $LD(M)$ consisting of derivations starting in $\mathbf{f}$ can be interpreted as a directed acyclic graph $G_\mathbf{f}$, where any element $(\mathbf{f}, t, h)$ of this set corresponds to a path from the root node $\mathbf{f}$ to the node t. Intermediate strings of source language trees or target language symbols form the nodes of this graph, the tree transducer productions in a derivation history are weighted edges. Finding the best derivation of $\mathbf{f}$ according to the model from section IV-D is thus eqivalent to finding the shortest path in $G_\mathbf{f}$ from the node $\mathbf{f}$ to a node $\mathbf{e} \in \Delta^*$.

For now we have implemented three algorithms that solve the shortest path problem approximately:

1) Greedy search (symbol: GREEDY);
2) A* search (symbol: ASTAR);
3) Stack decoding (symbol: STACK$[n]$, where n is the size of the stacks).

Here stack decoding is a variant of the algorithm successfully used for phrase-based decoding in PHARAOH [21] and later in MOSES [22]. Originally there are as many stacks as words in a source sentence and a translation hypothesis is put into the k-th stack if it covers k source language words. Therefore the last stack will contain only complete translations. We modified the algorithm to allow for translation hypotheses that do not cover new words compared to their predecessor. This happens if a rule does not contain any terminal symbols in its source language part, e.g. rules that only perform re-orderings of subtrees or rules that translate the phrase compositionally. The original algorithm would have attempted to put a hypothesis resulting from such a rule into the same stack as its predecessor. Due to monotonicity, the new hypothesis has necessarily a higher cost than its source and might be pruned from the stack. In our version of stack decoding there is a second dimension of stacks for each word that gets extended whenever it is needed. That way a hypothesis is never put into the same stack as its predecessor.

All three algorithms require a heuristic function that guides the search through the graph. We adapted the heuristic function for phrase-based translation from [21] in the following way: For all subtrees we choose the cheapest lexical[6] rule and calculate its costs based on the rule features and the language model. The language model cost for the target side of the rule is calculated in isolation from other lexcial rules. For all trees or subtrees for which no lexical rules exist, we search for the cheapest cover with lexical rules of their subtrees. Next we multiply the costs of all trees in the cover. Unfortunately such a heuristic function is not necessarily admissible. In the case of A* search this means that the search will not be optimal.

All algorithms that build their search graph based on the set of left-most derivations $LD(M)$ traverse the input trees from left to right and in a top-down manner. Bottom-up approaches to decoding algorithms for syntax-based translation have been presented in [14] and [15]. We plan to implement variants of these in the next versions of BONSAI.

VII. PRELIMINARY EVALUATION

The baseline system used for comparison is the phrase-based SMT package MOSES [22]. MOSES as well as BONSAI have been trained on the Polish-French part of the JRC-Acquis parallel corpus from which 2000 sentences were set apart and divided equally into development set and test set. The training procedure for both systems is identical until the actual extraction of phrases and rules starts. The development set was used for minimal-error-rate training (MERT) to create a configuration with optimized translation model weights for

[6]A lexical rule does not contain any non-terminal symbols.

TABLE I
SOME EVALUATION RESULTS FOR DIFFERENT CONFIGURATIONS

Algorithm	$\bar{t}$ [s]	BLEU	METEOR
MOSES DEFAULT		0.5906	0.3944
MOSES MERT		0.6259	0.4340
BONSAI GREEDY	0.05	0.6022	0.3972
BONSAI ASTAR	1.33	0.6104	0.4123
BONSAI STACK[5]	0.20	0.6140	0.4067
BONSAI STACK[10]	0.35	0.6166	0.4122
BONSAI STACK[20]	0.63	0.6194	0.4133
BONSAI STACK[50]	1.45	0.6234	0.4188
BONSAI STACK[100]	2.74	0.6247	0.4194

MOSES. This configuration is denoted by the symbol MOSES MERT.

For BONSAI we have not constructed a procedure for parameter optimization yet, although it should not be a problem to adapt MERT for this goal once the main programming work is finished. Instead we use weights that were set after a few manual trials. Therefore we use a second baseline system MOSES DEFAULT, equal to MOSES MERT but with default parameters instead of optimized weights. We evaluated the translation results of MOSES using its standard settings, i.e. its stack decoding algorithm uses a stack size of 100. Our system is denoted by the symbol BONSAI combined with the symbols of the search methods introduced in the previous section.

The evaluation results have been compiled in Table I. All configurations have been evaluated using the BLEU and METEOR metrics. Concerning the BLEU scores, BONSAI fares better than MOSES DEFAULT for all configurations and comes very close to the performance of MOSES MERT when the stack-decoding algorithm with greater stack sizes[7] is used. The results are slightly worse for the METEOR metric when compared to MOSES MERT, but MOSES DEFAULT is again surpassed in all cases.

These—and especially the scores of BONSAI STACK[100]—are very promising results, all the more so, if we remember that the model parameters of BONSAI where not optimized by an appropriate training procedure and that the parser used is only a shallow parser that may produce a lot of parse errors. Despite its crude implementation, BONSAI seems to be on par with a state-of-the-art phrase-based SMT system and there is still plenty of room for improvements.

Along with the translation quality for the algorithms implemented in BONSAI, we compare the average time $\bar{t}$ taken for the translation of one sentence from the test set. High processing speed is especially important in the context of commercial MT systems. Here stack decoding with smaller stack sizes seems to be a reasonable compromise between quality and speed. We hope however that we will be able to improve the speed in future version.

[7]Note however that we do not use stack sizes greater than than 100 which is also the stack size used for MOSES.

VIII. DISCUSSION AND FUTURE WORK

The syntax-based SMT approaches based on tree transducers bear a strong resemblance to the rule-based syntactic transfer approach of TRANSLATICA presented in section II. Both methods require the input sentence to be parsed prior to translation. In both cases translation is achieved by tree transformations: reordering, insertion and deletion of subtrees and the translation of leave symbols. TRANSLATICA performs a depth-first left-to-right postorder traversal (subtrees are reordered after processing) of input trees. At the moment the presented SMT model is a close implementation of **xRs** as introduced by [11] and therefore the processing direction is top-down in a depth-first left-to-right preorder fashion (subtrees are reordered before processing), but this is due to change in the near future in favour of an equivalent post-order algorithm. Such an approaches have been put forward by, for instance, [15].

Leaving aside formal aspects of the expressive power of the language used for transfer rule programming, it seems plausible to interpret the syntactic transfer implemented in TRANSLATICA as a deterministic tree transducer. The tree or tree-to-string transducers described in section III and consist of millions of rules and are obviously non-deterministic, since they most certainly contain productions that have the same source language side and different target language sides. The same is true for BONSAI.

Once the processing order of BONSAI has been changed to a postorder approach, a combination of both methods on the level of subtrees of the same input tree seems desirable. Both systems, TRANSLATICA and BONSAI, process subtrees independently from their siblings, and it is possible to let them work in parallel. Using the search methods incorporated in BONSAI would also allow for a limited non-determinism in TRANSLATICA which could now provide all possible translations for a found lexicon entry that in turn could be disambiguated by BONSAI's language model. On the other hand, the semantic information encoded in *Translatica*, its handling of verb frames, and its capabilities to handle general context-independent translations would make it a relieable back-bone for the highly context-dependent translations of BONSAI. One can imagine a general rule that BONSAI is only allowed to translate a given subtree if a rule matches at least one non-terminal symbol while unknown constructions or compositional translations are left to TRANSLATICA.

Nevertheless, before we can approach a combination of TRANSLATICA and BONSAI, the following matters (among others) need to be examined and solved:

The influence that parsing quality has on syntax-based SMT needs to be investigated. In this paper we have presented a syntax-based system that relies on the results of a shallow parser and yet is on par with a modern phrase-based system. It has to be seen if the use of a carefully developed deep parser brings any improvements. We do not know of any work in syntax-based SMT that has compared the impact of different parsers within the same SMT system.

The most work in syntax-based SMT focuses on languages that are weakly inflectional and have a strict word order. So it remains to investigate how complex non-terminal symbols that contain additional features affect translation quality. For Polish to French translation it could be helpful to encode information whether the noun phrases attached to a sentence are verb complements, modifiers, or the subject, since this does not follow necessarily from their order in the sentence.

We started the paper with the observation that—apart from one example—no commercial MT system that is usable on a stand-alone desktop or notebook computer actually seems to be using statistical techniques for translation, let alone methods from syntax-based SMT. This may be due to the amount of resources required. In section V it became obvious that a compact and efficient storage scheme needs to be developed, allowing access to the millions of translation rules created during training. Compression however is only one part of the solution, it is also necessary to minimize the number of rules necessary for high quality translation. We believe that a future combination with a rule-based MT system might also be helpful in this respect.

References

[1] J. Hutchins, "Compendium of translation software—directory of commercial machine translation systems and computer-aided translation support tools," Jan. 2009. [Online]. Available: http://www.hutchinsweb.me.uk/Compendium.htm

[2] Y. Zhang, S. Vogel, and A. Waibel, "Interpreting bleu/nist scores: How much improvement do we need to have a better system," in *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, 2004, pp. 2051–2054.

[3] K. Jassem, *Przetwarzanie tekstów polskich w systemie tłumaczenia automatecznego POLENG*. Poznań: Wydawnictwo Naukowe UAM, 2006.

[4] F. Graliński, "Some methods of describing discontinuity in polish and their cost-effectiveness," *Lecture Notes in Artificial Intelligence, Text, Speech and Dialogue, 9th International Conference, TSD 2006*, vol. 4188, 2006.

[5] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 868–876.

[6] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proceedings of the 39th Annual Meeting of the ACL*, 2001.

[7] D. Chiang, "A hierarchical phrase-based model for statistical machine translation." in *ACL*. The Association for Computer Linguistics, 2005.

[8] P. Koehn, F. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[9] A. Zollmann and A. Venugopal, "Syntax augmented machine translation via chart parsing," 2006.

[10] A. V. Aho and J. D. Ullman, "Translations on a context-free grammar," *Information and Control*, vol. 19, no. 5, pp. 439–475, December 1971.

[11] J. Graehl, K. Knight, and J. May, "Training tree transducers." *Computational Linguistics*, vol. 34, no. 3, pp. 391–427, 2008.

[12] J. Graehl and K. Knight, "Training tree transducers." in *HLT-NAACL*, 2004, pp. 105–112.

[13] J. Eisner, "Learning non-isomorphic tree mappings for machine translation," in *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 205–208.

[14] L. Huang, "Statistical syntax-directed translation with extended domain of locality," in *In Proc. AMTA 2006*, 2006, pp. 66–73.

[15] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation." in *ACL*. The Association for Computer Linguistics, 2006.

[16] A. Przepiórkowski and A. Buczyński, "♠: Shallow parsing and disambiguation engine," in *Proceedings of the 3rd Language & Technology Conference*, Poznań, 2007.

[17] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models." *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[18] ——, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[19] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga, "The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages," *CoRR*, vol. abs/cs/0609058, 2006, informal publication.

[20] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule," Proceedings of the Human Language Technology and North American chapter of the Association for Computational Linguistics Conference (HLT-NAACL,), May 2-5, Boston, MA, 2004.

[21] P. Koehn, "Pharaoh: A beam search decoder for phrase-based statistical machine translation models," in *Proceedings of AMTA 2004*, Washington, District of Columbia, 2004, pp. 115–124.

[22] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation." in *ACL*. The Association for Computer Linguistics, 2007.