Dependency-Based Statistical Machine Translation

Heidi J. Fox
Brown Laboratory for Linguistic Information Processing
Brown University, Box 1910, Providence, RI 02912
hjf@cs.brown.edu

Abstract

We present a Czech-English statistical machine translation system which performs tree-to-tree translation of dependency structures. The only bilingual resource required is a sentence-aligned parallel corpus. All other resources are monolingual. We also refer to an evaluation method and plan to compare our system's output with a benchmark system.

1 Introduction

The goal of statistical machine translation (SMT) is to develop mathematical models of the translation process whose parameters can be automatically estimated from a parallel corpus. Given a string of foreign words **F**, we seek to find the English string **E** which is a "correct" translation of the foreign string. The first work on SMT done at IBM (Brown et al., 1990; Brown et al., 1992; Brown et al., 1993; Berger et al., 1994), used a noisy-channel model, resulting in what Brown et al. (1993) call "the Fundamental Equation of Machine Translation":

$$\hat{\mathbf{E}} = \overset{\text{argmax}}{\mathbf{E}} \; P(\mathbf{E})P(\mathbf{F} \mid \mathbf{E}) \qquad (1)$$

In this equation we see that the translation problem is factored into two subproblems. $P(\mathbf{E})$ is the *language model* and $P(\mathbf{F} \mid \mathbf{E})$ is the *translation model*. The work described here focuses on developing improvements to the translation model.

While the IBM work was groundbreaking, it was also deficient in several ways. Their model translates words in isolation, and the component which accounts for word order differences between languages is based on linear position in the sentence. Conspicuously absent is all but the most elementary use of syntactic information. Several researchers have subsequently formulated models which incorporate the intuition that syntactically close constituents tend to stay close across languages. Below are descriptions of some of these different methods of integrating syntax.

- Stochastic Inversion Transduction Grammars (Wu and Wong, 1998): This formalism uses a grammar for English and from it derives a possible grammar for the foreign language. This derivation includes adding productions where the order of the RHS is reversed from the ordering of the English.

- Syntax-based Statistical Translation (Yamada and Knight, 2001): This model extends the above by allowing all possible permutations of the RHS of the English rules.

- Statistical Phrase-based Translation (Koehn et al., 2003): Here "phrase-based" means "subsequence-based", as there is no guarantee that the phrases learned by the model will have any relation to what we would think of as syntactic phrases.

- Dependency-based Translation (Čmejrek et al., 2003): This model assumes a dependency parser for the foreign language. The syntactic structure and labels are preserved during translation. Transfer is purely lexical. A generator builds an English sentence out of the structure, labels, and translated words.

Proceedings of the ACL Student Research Workshop, pages 91–96,
Ann Arbor, Michigan, June 2005.

2 System Overview

The basic framework of our system is quite similar to that of Čmejrek et al. (2003) (we reuse many of their ancillary modules). The difference is in how we use the dependency structures. Čmejrek et al. only translate the lexical items. The dependency structure and any features on the nodes are preserved and all other processing is left to the generator. In addition to lexical translation, our system models structural changes and changes to feature values, for although dependency structures are fairly well preserved across languages (Fox, 2002), there are certainly many instances where the structure must be modified.

While the entire translation system is too large to discuss in detail here, I will provide brief descriptions of ancillary components. References are provided, where available, for those who want more information.

2.1 Corpus Preparation

Our parallel Czech-English corpus is comprised of Wall Street Journal articles from 1989. The English data is from the University of Pennsylvania Treebank (Marcus et al., 1993; Marcus et al., 1994). The Czech translations of these articles are provided as part of the Prague Dependency Treebank (PDT) (Böhmová et al., 2001). In order to learn the parameters for our model, we must first create aligned dependency structures for the sentence pairs in our corpus. This process begins with the building of dependency structures.

Since Czech is a highly inflected language, morphological tagging is extremely helpful for downstream processing. We generate the tags using the system described in (Hajič and Hladká, 1998). The tagged sentences are parsed by the Charniak parser, this time trained on Czech data from the PDT. The resulting phrase structures are converted to tectogrammatical dependency structures via the procedure documented in (Böhmová, 2001). Under this formalism, function words are deleted and any information contained in them is preserved in features attached to the remaining nodes. Finally, functors (such as agent or patient) are automatically assigned to nodes in the tree (Žabokrtský et al., 2002).

On the English side, the process is simpler. We parse with the Charniak parser (Charniak, 2000) and convert the resulting phrase-structure trees to a function-argument formalism, which, like the tectogrammatic formalism, removes function words. This conversion is accomplished via deterministic application of approximately 20 rules.

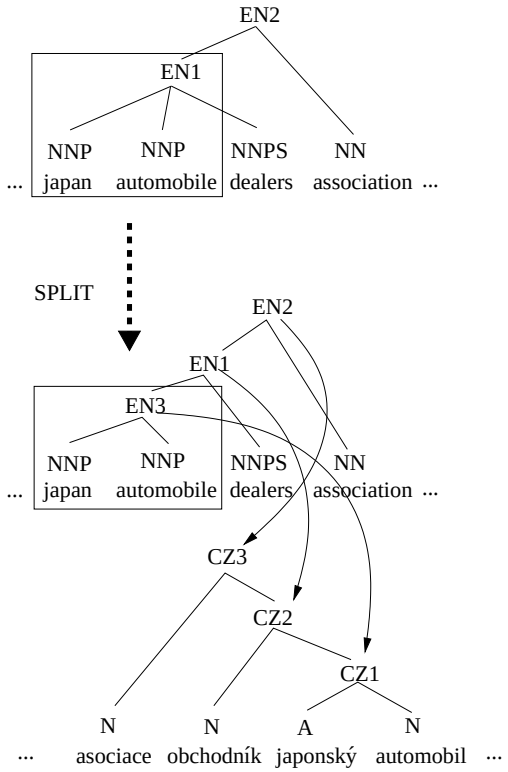

Figure 1: Left SPLIT Example

2.2 Aligning the Dependency Structures

We now generate the alignments between the pairs of dependency structures we have created. We begin by producing word alignments with a model very similar to that of IBM Model 4 (Brown et al., 1993). We keep fifty possible alignments and require that each word has at least two possible alignments. We then align phrases based on the alignments of the words in each phrase span. If there is no satisfactory alignment, then we allow for structural mutations. The probabilities for these mutations are refined via another round of alignment. The structural mutations allowed are described below. Examples are shown in phrase-structure format rather than dependency format for ease of explanation.

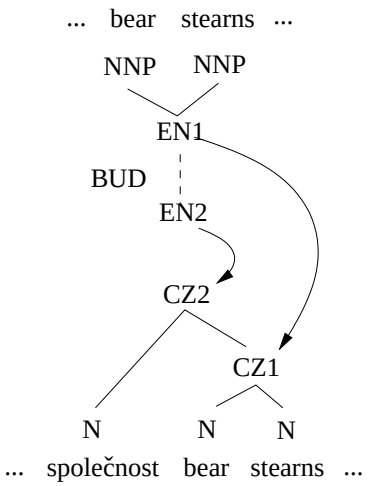

Figure 2: BUD Example

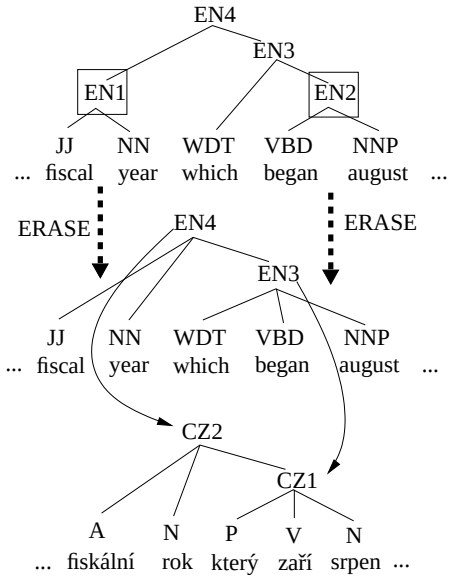

Figure 3: ERASE Example

- KEEP: No change. This is the default.
- SPLIT: One English phrase aligns with two Czech phrases and splitting the English phrase results in a better alignment. There are three types of split (left, right, middle) whose probabilities are also estimated. In the original structure of Figure 1, English node EN1 would align with Czech nodes CZ1 and CZ2. Splitting the English by adding child node EN3 results in a better alignment.
- BUD: This adds a unary level in the English tree in the case when one English node aligns to two Czech nodes, but one of the Czech nodes is the parent of the other. In Figure 2, the Czech has one extra word "společnost" ("company") compared with the English. English node EN1 would normally align to both CZ1 and CZ2. Adding a unary node EN2 to the English results in a better alignment.
- ERASE: There is no corresponding Czech node for the English one. In Figure 3, the English has two nodes, EN1 and EN2, which have no corresponding Czech nodes. Erasing them brings the Czech and English structures into alignment.
- PHRASE-TO-WORD: An entire English phrase aligns with one Czech word. This operates similarly to ERASE.

3 Translation Model

Given $\mathcal{E}$, the parse of the English string, our translation model can be formalized as $P(F \mid \mathcal{E})$. Let $\mathcal{E}_1 \ldots \mathcal{E}_n$ be the nodes in the English parse, $\mathcal{F}$ be a parse of the Czech string, and $\mathcal{F}_1 \ldots \mathcal{F}_m$ be the nodes in the Czech parse. Then,

$$P(F \mid \mathcal{E}) = \sum_{\mathcal{F} for F} P(\mathcal{F}_1 \ldots \mathcal{F}_m \mid \mathcal{E}_1 \ldots \mathcal{E}_n) \quad (2)$$

We initially make several strong independence assumptions which we hope to eventually weaken. The first is that each Czech parse node is generated independently of every other one. Further, we specify that each English parse node generates exactly one (possibly NULL) Czech parse node.

$$P(\mathcal{F} \mid \mathcal{E}) = \prod_{\mathcal{F}_i \in \mathcal{F}} P(\mathcal{F}_i \mid \mathcal{E}_1 \ldots \mathcal{E}_n) = \prod_{i=1}^{n} P(\mathcal{F}_i \mid \mathcal{E}_i) \quad (3)$$

An English parse node $\mathcal{E}_i$ contains the following information:

- An English word: e_i
- A part of speech: t_i^e
- A vector of n features (e.g. negation or tense): $< \phi_i^e[1], \ldots, \phi_i^e[n] >$

- A list of dependent nodes

In order to produce a Czech parse node $\mathcal{F}_i$, we must generate the following:

Lemma f_i: We generate the Czech lemma f_i dependent only on the English word e_i.

Part of Speech t_i^f: We generate Czech part of speech t_i^f dependent on the part of speech of the Czech parent $t_{par(i)}^f$ and the corresponding English part of speech t_i^e.

Features $< \phi_i^f[1], \ldots, \phi_i^f[n] >$: There are several features (see Table 1) associated with each parse node. Of these, all except IND are typical morphological and analytical features. IND (indicator) is a loosely-specified feature comprised of functors, where assigned, and other words or small phrases (often prepositions) which are attached to the node and indicate something about the node's function in the sentence. (e.g. an IND of "at" could indicate a locative function). We generate each Czech feature $\phi_i^f[j]$ dependent only on its corresponding English feature $\phi_i^e[j]$.

Head Position h_i: When an English word is aligned to the head of a Czech phrase, the English word is typically also the head of its respective phrase. But, this is not always the case, so we model the probability that the English head will be aligned to either the Czech head or to one of its children. To simplify, we set the probability for each particular child being the head to be uniform in the number of children. The head position is generated independent of the rest of the sentence.

Structural Mutation m_i: Dependency structures are fairly well preserved across languages, but there are cases when the structures need to be modified. Section 2.2 contains descriptions of the different structural changes which we model. The mutation type is generated independent of the rest of the sentence.

Feature	Description
NEG	Negation
STY	Style (e.g. statement, question)
QUO	Is node part of a quoted expression?
MD	Modal verb associated with node
TEN	Tense (past, present, future)
MOOD	Mood (infinitive, perfect, progressive)
CONJ	Is node part of a conjoined expression?
IND	Indicator

Table 1: Features

3.1 Model with Independence Assumptions

With all of the independence assumptions described above, the translation model becomes:

$$P(\mathcal{F}_i \mid \mathcal{E}_i) = P(f_i \mid e_i) P(t_i^f \mid t_i^e, t_{par(i)}^f) \times P(h_i) P(m_i) \prod_{j=1}^{n} P(\phi_i^f[j] \mid \phi_i^e[j]) \quad (4)$$

4 Training

The Czech and English data are preprocessed (see Section 2.1) and the resulting dependency structures are aligned (see Section 2.2). We estimate the model parameters from this aligned data by maximum likelihood estimation. In addition, we gather the inverse probabilities $P(E \mid F)$ for use in the figure of merit which guides the decoder's search.

5 Decoding

Given a Czech sentence to translate, we first process it as described in Section 2.1. The resulting dependency structure is the input to the decoder. The decoder itself is a best-first decoder whose priority queue holds partially-constructed English nodes.

For our figure of merit to guide the search, we use the probability $P(E \mid F)$. We normalize this using the *perplexity* (2^H) to compensate for the different number of possible values for the features $\phi[j]$. Given two different features whose values have the same probability, the figure of merit for the feature with the greater uncertainty will be boosted. This prevents features with few possible values from monopolizing the search at the expense of the other features. Thus, for feature $\phi_i^e[j]$, the figure of merit is

$$FOM = P(\phi_i^e[j] \mid \phi_i^f[j]) \times 2^{H(\Phi_i^e[j] | \phi_i^f[j])} \quad (5)$$

Since our goal is to build a forest of partial translations, we translate each Czech dependency node independently of the others. (As more conditioning factors are added in the future, we will instead translate small subtrees rather than single nodes.) Each translated node $\mathcal{E}_i$ is constructed incrementally in the following order:

1. Choose the head position h_i
2. Generate the part of speech t_i^e
3. For $j = 1..n$, generate $\phi_i^e[j]$
4. Choose a structural mutation m_i

English nodes continue to be generated until either the queue or some other stopping condition is reached (e.g. having a certain number of possible translations for each Czech node). After stopping, we are left with a forest of English dependency nodes or subtrees.

6 Language Model

We use a syntax-based language model which was originally developed for use in speech recognition (Charniak, 2001) and later adapted to work with a syntax-based machine translation system (Charniak et al., 2001). This language model requires a forest of partial phrase structures as input. Therefore, the format of the output of the decoder must be changed. This is the inverse transformation of the one performed during corpus preparation. We accomplish this with a statistical tree transformation model whose parameters are estimated during the corpus preparation phase.

7 Evaluation

We propose to evaluate system performance with version 0.9 of the NIST automated scorer (NIST, 2002), which is a modification of the BLEU system (Papineni et al., 2001). BLEU calculates a score based on a weighted sum of the counts of matching n-grams, along with a penalty for a significant difference in length between the system output and the reference translation closest in length. Experiments have shown a high degree of correlation between BLEU score and the translation quality judgments of humans. The most interesting difference in the NIST scorer is that it weights n-grams based on a notion of informativeness. Details of the scorer can be found in their paper.

For our experiments, we propose to use the data from the PDT, which has already been segmented into training, held out (or development test), and evaluation sets. As a baseline, we will run the GIZA++ implementation of IBM's Model 4 translation algorithm under the same training conditions as our own system (Al-Onaizan et al., 1999; Och and Ney, 2000; Germann et al., 2001).

8 Future Work

Our first priority is to complete the final pieces so that we have an end-to-end system to experiment with. Once we are able to evaluate our system output, our first priority will be to analyze the system errors and adjust the model accordingly. We recognize that our independence assumptions are generally too strong, and improving them is a hight priority. Adding more conditioning factors should improve the quality of the decoder output as well as reducing the amount of probability mass lost on structures which are not well formed. With this will come sparse data issues, so it will also be important for us to incorporate smoothing into the model.

There are many interesting subproblems which deserve attention and we hope to examine at least a couple of these in the near future. Among these are discontinuous constituents, head switching, phrasal translation, English word stemming, and improved modeling of structural changes.

Acknowledgments

This work was supported in part by NSF grant IGERT-9870676. We would like to thank Jan Hajič, Martin Čmejrek, Jan Cuřín for all of their assistance.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation: Final report, JHU workshop 1999. www.clsp.jhu.edu/ws99/projects/mt/final_report/mt-final-report.ps.

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer, Harry Printz, and Luboš Ureš. 1994. The Candide system for machine translation. In *Proceedings of the ARPA Human Language Technology Workshop*.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2001. The Prague Dependency Treebank: Three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

Alena Böhmová. 2001. Automatic procedures in tectogrammatical tagging. *The Prague Bulletin of Mathematical Linguistics*, 76.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Peter F. Brown, Stephen A. Della Petra, Vincent J. Della Pietra, John D. Lafferty, and Robert L. Mercer. 1992. Analysis, statistical transfer, and synthesis in machine translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 83–100.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2001. Syntax-based language models for statistical machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, July.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 116–123, Toulouse, France, July.

Martin Čmejrek, Jan Cuřín, and Jiří Havelka. 2003. Czech-English Dependency-based Machine Translation. In *EACL 2003 Proceedings of the Conference*, pages 83–90, April 12–17, 2003.

Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, July.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

Jan Hajič and Barbora Hladká. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of COLING-ACL Conference*, pages 483–490, Montreal, Canada.

Philip Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Edmonton, Canada, May.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 13(2):313–330, June.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 114–119.

NIST. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. Technical report, IBM.

Dekai Wu and Hongsing Wong. 1998. Machine translation with a stochastic grammatical channel. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 1408–1414.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

Zdeněk Žabokrtský, Petr Sgall, and Sašo Džeroski. 2002. Machine learning approach to automatic functor assignment in the Prague Dependency Treebank. In *Proceedings of LREC 2002 (Third International Conference on Language Resources and Evaluation)*, volume V, pages 1513–1520, Las Palmas de Gran Canaria, Spain.