

Exploiting Syntactic Relationships in a Phrase-based Decoder: An Exploration

Tim Hunter and Philip Resnik*
University of Maryland

Abstract.

Phrase-based decoding is conceptually simple and straightforward to implement, at the cost of drastically oversimplified reordering models. Syntactically aware models make it possible to capture linguistically relevant relationships in order to improve word order, but they can be more complex to implement and optimise.

In this paper, we explore a new middle ground between phrase-based and syntactically informed statistical MT, in the form of a model that supplements conventional, non-hierarchical phrase-based techniques with linguistically informed reordering based on syntactic dependency trees. The key idea is to exploit linguistically-informed hierarchical structures only for those dependencies that cannot be captured within a single flat phrase. For very local dependencies we leverage the success of conventional phrase-based approaches, which provide a sequence of target-language words appropriately ordered and ready-made with any agreement morphology.

Working with dependency trees rather than constituency trees allows us to take advantage of the flexibility of phrase-based systems to treat non-constituent fragments as phrases. We do impose a requirement — that the fragment be a novel sort of “dependency constituent” — on what can be translated as a phrase, but this is much weaker than the requirement that phrases be traditional linguistic constituents, which has often proven too restrictive in MT systems.

Keywords: statistical MT, phrase-based translation, syntax, reordering

1. Introduction

A significant step forward in machine translation was the move from word-based translation models, originating from the IBM approach (Brown et al., 1990), to phrase-based translation models. This allows a sequence of contiguous source-language words to be atomically replaced with a sequence of contiguous target-language words. These “phrases” need not be phrases in the sense of any pre-existing theory of natural

* We thank Chris Dyer and Adam Lopez for many helpful comments and discussions, and Jeremy Kahn for the use of his EDPM evaluation software (which, in turn, uses the parser developed by Eugene Charniak and Mark Johnson). This work has been supported in part by Department of Defense contract RD-02-5700 and the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-02-001. Any opinions, findings, conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsors.



language syntax. Indeed the freedom to translate fragments such as ‘he said’ as a unit, despite the fact that this is not normally treated as a phrase in syntactic theories, is considered a strength of this approach.

Another natural modification of the IBM models is to arrange (the target-language sides of) the translation units in a tree structure rather than in a flat linear structure, in an attempt to more accurately model long-distance dependencies and cross-linguistic variation in word order. The choice of flat structures or tree structures is independent of the choice of words or phrases as translation units.

A more recent step forward in statistical machine translation, hierarchical phrase-based models, integrates these two variations on the IBM approach, using multi-word translation units arranged in a tree structure. In addition, however, these systems allow phrases to be nested within each other, and therefore use translation units which are themselves hierarchical, unlike the original phrase-based systems.

The alternative suggested here arranges multi-word translation units in hierarchical structures (specifically, dependency structures), but the translation units themselves have no hierarchical structure: they are exactly those used in the original phrase-based systems. In this respect it explores a kind of “middle ground” between existing approaches. As with all syntax-based approaches, we use structural relations to judge the goodness of an ordering of (the target-language sides of) translation units in a somewhat linguistically-informed way; more specifically, we train a model of the target language’s word order using a dependency corpus, and then use this model to judge the goodness of a particular ordering of target-language phrases. In contrast, in the original phrase-based systems, the only well-defined question to ask about a particular ordering of phrases is to what extent it differs from a direct monotonic translation (the “default case”, carried over from the speech recognition systems which inspired the IBM models).

Although it must be stated at the outset that the approach we describe failed to produce improvements over a conventional phrase-based MT baseline, we identify two contributions that this paper makes. The first is the description of a particular model integrating dependency structures with flat phrase translation units via a tree-flattening mechanism, and the decoding algorithm that arises from it. The second is its value as a case study, which we hope might be useful for other researchers, of the development of some intuitively appealing ideas into an explicit model and decoding algorithm to the point that permits experimental evaluation, and clear-eyed assessment of the results; in particular, we note evidence that dependency structures may not encode source-target commonalities as faithfully as intuition suggests.

The rest of this paper is organised as follows. In Section 2 we review in more detail previous approaches to statistical machine translation and how they relate to the system we propose here. We present in detail our dependency-based reordering model in Section 3, and the accompanying decoding algorithm in Section 4.¹ Finally we discuss results of experiments translating Czech to English and Arabic to English in Section 5 and conclude in Section 6.

2. Previous Related Work

In this section we briefly review other approaches to integrating syntactic information with phrase-based translation.

In the original phrase-based translation systems (Och et al., 1999; Koehn et al., 2003), we can identify two ways in which “reorderings” of sentence fragments, be they words or larger fragments, can occur. First, the two reordered pieces might be covered by a single phrase. For example, we might have extracted $\langle \textit{ballon rouge}, \textit{red ball} \rangle$ as a translation unit during training. In this case the reordering “comes for free”. The second, more complicated case, is where the two reordering pieces are not covered by a single phrase. To generate correct translations of this sort, the decoder must decide to reverse the order of the two separate phrases. For example, if our extracted phrases include only $\langle \textit{rouge}, \textit{red} \rangle$ and $\langle \textit{ballon}, \textit{ball} \rangle$ and we wish to translate *ballon rouge*, the decoder must decide to prefer the order *red ball* over *ball red*. Standard phrase-based translation systems simply set a preference for “monotonic” orderings of phrases, and rely on other models (eg. language models) to override this preference when appropriate.

Another line of work, originally independent of the shift from word-based to phrase-based translation systems, sought to take advantage of cross-linguistic generalisations concerning variation in word order. In general this requires arranging the words of a sentence in some sort of *hierarchical* representation of the structure of sentences, in contrast to the phrase-based approaches described above. Yamada and Knight (2001), for example, parse the source sentence to be translated and apply a statistical model to choose suitable reorderings. For each node of the parse tree, they use a probability distribution over the $n!$ different orderings of the node’s n children; word-to-word translation applies at each terminal node of the resulting reordered tree (with the additional possibility that nodes may be inserted). This “reorder-insert-translate” noisy channel model is used at decode-time. Similar strategies, where hierarchical structures of words are evaluated for linguistic “goodness” at decode-time, were adopted by Wu and Wong (1998), Gildea (2003)

and Galley et al. (2004), among others. Another approach, developed by Collins et al. (2005) and Xia and McCord (2004), is to use hierarchical structures to reorder the source sentence as a preprocessing step, and feed the reordered sentence to a standard phrase-based decoder. The intuition behind this is that if all the significant reorderings are taken care of in the preprocessing, then the phrase-based system’s bias towards monotonic decoding will be well-suited to translating the result.²

These two advances — translating multiple words atomically, and linguistically evaluating hierarchical arrangements of words — were integrated in a number of systems that arranged multi-word translation units in hierarchical structures. Chiang (2005) generalises the training procedure to extract not only pairs of flat phrases like $\langle \textit{ballon rouge, red ball} \rangle$, but also pairs of sequences that may contain “variables” (or, understood as part of a synchronous context-free grammar, non-terminals), such as $\langle \textit{ne X pas, not X} \rangle$. These rules are not based on any parse trees, but rather are extracted from an aligned parallel corpus on the basis of co-occurrence; the resulting system is therefore, as Chiang notes, “formally syntax-based” but not “linguistically syntax-based”. Quirk et al. (2005) adopt a similar but linguistically syntax-based approach using dependency trees, extracting treelet-pairs from a parallel corpus with dependency parses on one side during training; Shen et al. (2008) integrate this dependency-treelet idea with a dependency language model that conditions on linear order somewhat similarly to the reordering model we present below. In these systems, once a selection of translation units has been decided upon that covers the source sentence to be translated, there is no independent question of how these translation units should be ordered; the order of *not* and the translation of *X*, in the simple example above, is determined by the target side of the translation unit. This distinguishes them from both flat phrase-based approaches and from earlier syntax-based approaches, where the choice of translation units was independent of the order in which the selected fragments of the target language will be arranged.

The approach we describe in this paper explores a middle ground between the three kinds of systems described above. We combine multi-word translation units with hierarchical notions. However, hierarchy is only introduced *among translation units*. Unlike Chiang (2005) and Quirk et al. (2005), our translation units are not hierarchical; they are exactly the ones used in the original phrase-based systems (Koehn et al. (2003) etc.). Rather than the purely monotonic-biased reordering model, we use hierarchical notions to decide among orderings of these flat phrases. These decisions are made as part of the decoding process, as for Yamada and Knight (2001) and others, not as a preprocessing step as for Collins et al. (2005) and Xia and McCord (2004). Like

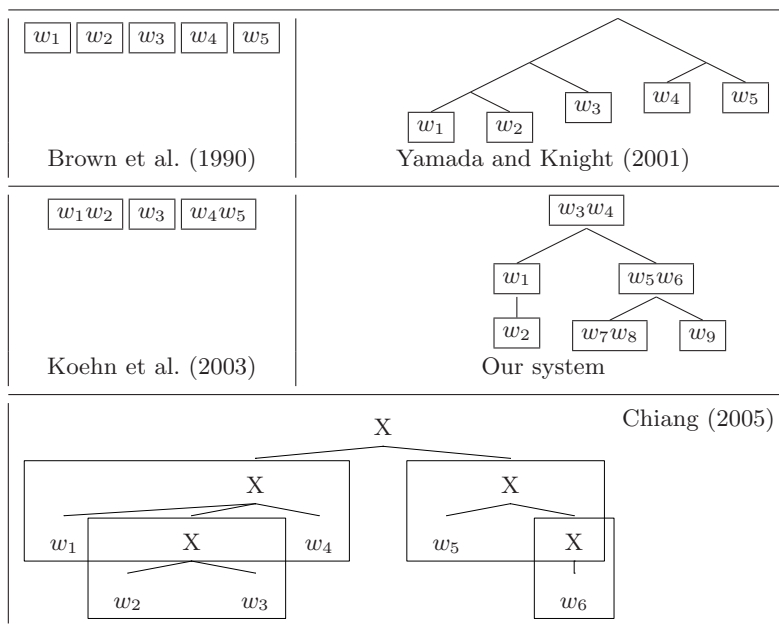


Figure 1. Various strategies for arranging translation units. Boxes represent translation units, and w_i represent words.

Quirk et al. (2005), but unlike Chiang (2005), we adopt a linguistically syntax-based approach, and also use dependency structures rather than constituency structures; this choice permits some additional flexibility with respect to the sense in which the two languages’ structures must “match”, as we will discuss below. Recent work presented in Galley and Manning (2008) is very similar to ours in that it builds hierarchical structures of flat phrases, but in contrast is formally syntax-based and not linguistically syntax-based.

A system that occupies this middle ground inherits a number of technical simplicities from the original phrase-based systems. Since we use only the conventional flat phrases as translation units, our system can use “phrase tables” designed for use in earlier systems. We also maintain a left-to-right decoding algorithm, avoiding the need for more complex tree-based decoding procedures. Our approach differs from a standard phrase-based system only in having an additional model contribute scores to hypotheses. In short, we aim to retain the benefits and simplicity of phrase-based translation, but replace the least appealing aspect of such systems — the bias towards monotonic translations — with a linguistically-informed reordering model.

The relationships between this new approach and various existing systems are illustrated in Figure 1 and Table I.

Table I. One way of segmenting the space of statistical machine translation systems

Translation units	Arranged linearly	Arranged hierarchically
one word	Brown et al. (1990)	Yamada and Knight (2001)
flat multi-word	Koehn et al. (2003)	Our system
hierarchical multi-word		Chiang (2005)

3. Model

As is common, the probability of an English translation e given a foreign sentence f is computed via a log-linear model:

$$P(e|f) = P_\phi(f|e)^{\lambda_\phi} \times P_\ell(e)^{\lambda_\ell} \times P_d(e|f)^{\lambda_d} \times \omega^{|e|\lambda_\omega} \quad (1)$$

where P_ϕ is the translation probability model, P_ℓ is the language model, P_d is the distortion model, and ω is the word penalty. Only the distortion model P_d differs from that used in standard phrase-based systems.

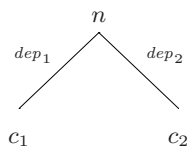
We use a monolingual English dependency corpus to train the ordering model P_d . In very broad terms, the decoding stage works as follows: we begin with a dependency tree for the foreign sentence to be translated, project the dependency structure to the English side, and use the trained ordering model to measure the goodness of a particular linearisation of the “English dependency tree”.

In order to clearly present the model we use for assigning scores to orderings of phrases in Section 3.2, we first introduce a model assigning scores to orderings of *words* in Section 3.1, which provides the basic intuition to be exploited.

3.1. BASIC INTUITIONS: WORD-REORDERING MODEL

Given a dependency-parsed corpus of (say) English, we can train a model of the probability (in English) of various linearisations of the nodes of any dependency tree. *Ignoring linearisations where dependency-subtrees are not contiguous*, choosing a linearisation amounts to choosing an ordering for each set $\{n, c_1, c_2, \dots, c_k\}$, where the c_i are the child subtrees of node n . We can approximate the probability of such an ordering O by assuming that the probability of a child subtree being at a particular linear offset from the parent n is independent of the offsets of its sisters from the shared parent n :

$$P(O) = \prod_{i=1}^k P(\delta_i | dep_i) \quad (2)$$



$$\begin{aligned}
 P(nc_1c_2) &= P(+1|dep_1) P(+2|dep_2) \\
 P(nc_2c_1) &= P(+1|dep_2) P(+2|dep_1) \\
 P(c_1nc_2) &= P(-1|dep_1) P(+1|dep_2) \\
 P(c_1c_2n) &= P(-2|dep_1) P(-1|dep_2) \\
 P(c_2nc_1) &= P(-1|dep_2) P(+1|dep_1) \\
 P(c_2c_1n) &= P(-2|dep_2) P(-1|dep_1)
 \end{aligned}$$

Figure 2. Scores for orderings of the nodes of a very simple dependency tree.

where dep_i is the label on the dependency linking (the head of) child subtree c_i to its parent n , and δ_i is the offset between n and (the closest edge of) c_i in ordering O , measured as a number of words. The probabilities on the right hand side of (2) can be easily estimated via maximum likelihood estimates from frequencies in a sufficiently-large dependency-parsed English corpus. Figure 2 illustrates probabilities for the six possible word orderings given a simple tree; e.g., in the ordering nc_1c_2 , word c_2 occurs “two steps to the right” of head n , linked via dependency dep_2 , hence the factor $P(+2|dep_2)$.

By assuming in addition that the probability of a particular ordering of each node and its child subtrees is independent of the ordering of any other node in the tree and its child subtrees, we can find the linearisation of the words in a foreign dependency-parsed sentence which most closely matches “English word order”, by choosing the best ordering O for each set containing a node and its child subtrees.

Given a table of translations for single foreign words, a naive translation method would be to take the most “English-like” ordering of the foreign words, and translate each foreign word individually. The next section integrates the motivation behind this naive approach with the advantages of phrase-based translation systems.

3.2. PHRASE-REORDERING MODEL

The reordering model P_d in (1) needs to assign a probability to a particular ordering of phrases, rather than words. To combine the intuitions behind the word-reordering model just discussed with the advantages of phrase-based translation systems, we aim to translate multi-word fragments atomically, and order the translations of these fragments according to the reordering model.

Suppose that we need to translate the foreign language sentence **MAN THE WOMAN TALL THE SAW** (call this sentence f , and assume it in fact means ‘the man saw the tall woman’) into English given the dependency tree in Figure 3. Before applying a reordering model to determine an English-like ordering of the nodes of the dependency tree, we can “compress” some portions of the dependency tree to produce flat, multi-word phrases. Four of the many possibilities for this dependency

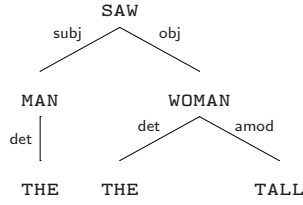


Figure 3. Dependency parse of the foreign sentence to be translated

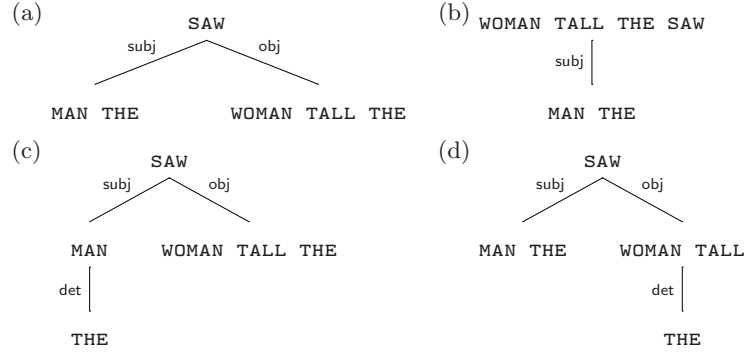


Figure 4. Four possible compressions of the dependency tree in Figure 3.

tree are shown in Figure 4. Details of the possible ways to compress dependency trees will be explained shortly (subsection 3.3).

Suppose we choose the compressed tree in Figure 4(c), and an English translation for each of the four foreign phrases in that tree; say, *saw* for **SAW**, *the* for **THE**, *guy* for **MAN**, and *the tall lady* for **WOMAN TALL THE**. These are taken from exactly the same kind of phrase table as is used in standard phrase-based systems. We must now choose how to order these four English phrases. The distortion scores for some of the possibilities are shown in (3). Since English subjects generally tend to be at a small negative (leftwards) offset from their governors (verbs), and English objects generally tend to be at a small positive (rightwards) offset from their governors (verbs), (3c) will probably be

$$\begin{aligned}
 P_d([saw][the\ tall\ lady][the][guy]|f) \\
 = P(+4|subj) P(+1|obj) P(-1|det)
 \end{aligned}
 \tag{3a}$$

$$\begin{aligned}
 P_d([the\ tall\ lady][saw][the][guy]|f) \\
 = P(+1|subj) P(-1|obj) P(-1|det)
 \end{aligned}
 \tag{3b}$$

$$\begin{aligned}
 P_d([the][guy][saw][the\ tall\ lady]|f) \\
 = P(-1|subj) P(+1|obj) P(-1|det)
 \end{aligned}
 \tag{3c}$$

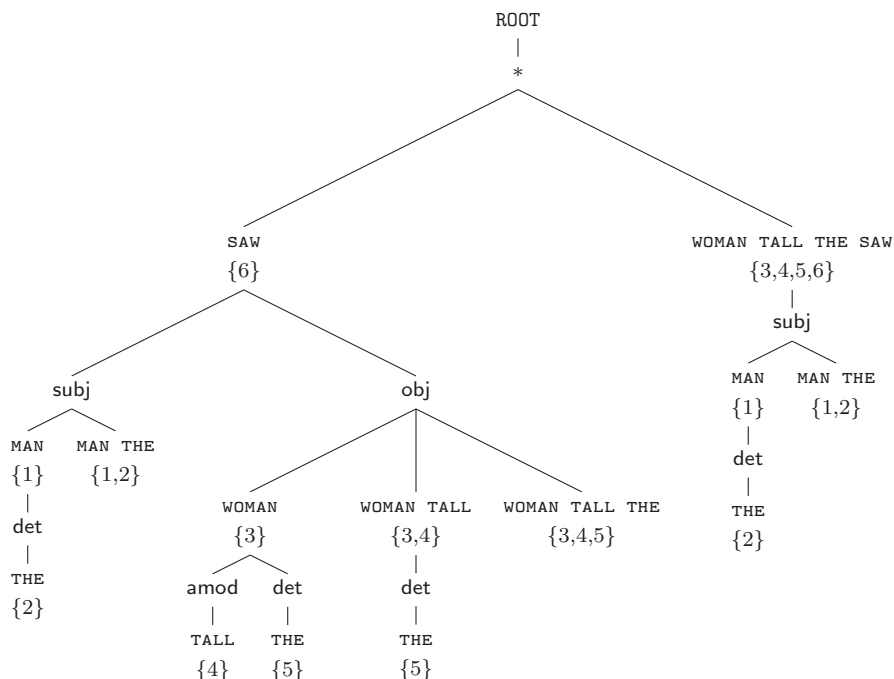


Figure 5. A representation of all the possible compressions of the dependency tree from Figure 3

the highest score of these three. Note that in (3a) we take the offset from the governor *saw* to the *closest word* of the *subj* dependent subtree, namely *the*, despite the fact that the head of this subtree is *MAN*.

3.3. THE ALLOWABLE COMPRESSIONS OF A TREE

Choosing a compression of a dependency tree rooted at a node n with child subtrees c_1, c_2, \dots, c_k amounts to choosing a set $C \subseteq \{c_1, c_2, \dots, c_k\}$. The chosen subtrees are the ones which are to be “squashed up” to combine with their parent node: the words contained in the subtrees $c_i \in C$, plus the word at node n , together form the phrase at the new root node n' . As children, n' has one compression of each of the $c_i \notin C$.

The choice of the set C is subject to two constraints on the phrase at n' (that is, the words contained in each of the subtrees $c_i \in C$ together with the word at node n): (i) the phrase must be contiguous in the original sentence, and (ii) the length of the phrase in words must not exceed a pre-specified threshold.

The total range of possible compressions of the dependency tree in Figure 3, with a phrase-length threshold of four (which is the value we used in our experiments), is represented in Figure 5. This tree can

be read somewhat like an “and-or tree”. The left-hand path at the first branch of the tree represents choosing to use [SAW] as one of the phrases to cover the input sentence with. From this point, we must find a way to cover both the subj dependent *and* the obj dependent. In order to cover the subj dependent we must use either [MAN] (and in turn [THE]), *or* [MAN THE]; in order to cover the obj dependent we must use either [WOMAN] *or* [WOMAN TALL] *or* [WOMAN TALL THE] (with further consequences in the first two cases). The two children of the node labelled * represent the two alternate ways of compressing the subtree linked by the * dependency to the ROOT node; that is, the two alternate ways of compressing the entire original dependency tree. Note that there is no compression of this subtree where WOMAN and THE have been combined into one phrase with TALL as a child, because WOMAN and THE are not adjacent in the original source sentence.

To translate the foreign sentence we must choose a subset of the “phrase nodes” of this graph which covers the sentence exactly. These phrase nodes will form a dependency tree like those in Figure 4. Completing a translation therefore means choosing an ordering of these foreign phrases and choosing an English translation for each of these phrases from the phrase table.

Given the set of n phrases we are going to use to cover the input sentence, not all of the $n!$ possible orderings of (the English translations of) these phrases are permissible. The reordering model imposes a further restriction: it only assigns a score to those linearisations where each subtree is linearly contiguous. For example, the selection of the phrases WOMAN TALL, THE, SAW and MAN THE from Figure 5 amounts to the selection of the compressed dependency tree in Figure 4(d). If these phrases were arranged linearly as, for example, [WOMAN TALL] [SAW] [THE] [MAN THE], the reordering model would not be able to assign a score to the obj dependency linking SAW and (the subtree headed by) WOMAN TALL, because there is no nearest edge of the subtree headed by WOMAN TALL; the subtree has been split. This imposes a non-trivial restriction on the range of phrases we can choose from when considering ways to extend a hypothesis during decoding; see Hunter and Resnik (2009) for further discussion of this and some other aspects of the model that we can discuss only briefly here.

3.4. SOME FURTHER DETAILS

3.4.1. *Computing offsets from complex governor phrases*

In the examples in (3), the governor phrase of each dependency was always a single word. Those examples used compressions where dependent phrases were longer than a single word, in which case we took the

closest edge of this phrase when calculating offsets. However, when a governor phrase is longer than a single word, it does not always make sense to determine offsets on the basis of the closest edge of the governor phrase. For example, consider the following two orderings of phrases:

$$P_d([\textit{saw the tall lady}][\textit{the guy}]) = ?$$

$$P_d([\textit{saw}][\textit{the tall lady}][\textit{the guy}]) = ?$$

We would like our model to assign the same score to the way we have linearised the `subj` dependency between `SAW` and `MAN` in each of these two cases, but if we only consider nearest edges, then the relevant offset for this dependency will be +1 in the first case but +4 in the second. The problem here is that in the first case, the governor phrase [*saw the tall lady*] itself contains (our translation of) the `obj` dependent of `SAW`, and the placement of this dependent is relevant to the linearisation of (our translation of) the `subj` dependent [*the guy*].

The solution we adopt is to calculate each dependency’s offset on the basis of the word in the target-side governor phrase which is aligned to the governor word in the uncompressed source dependency tree. In the example above, this means we take the offset from the word *saw* to the phrase [*the guy*], because this is the word which is aligned to the source-language word which is the “real” governor of the `subj` dependency in the original uncompressed tree (Figure 3), namely `SAW`. This requires that the entries in our phrase table come with word alignments: otherwise there is no sense in which we can identify the English word *saw* as the “head word” of the English phrase [*saw the tall lady*], because there is no uncompressed English dependency tree.

3.4.2. Normalisation of dependency labels

As presented, this model assumes that there is a desirable target-language translation of the input sentence which shares not only the same dependency structure (modulo some flattening), but also the same *labels* on the dependencies. To allow for the possibility that a dependency labelled, say, `subj` in the input sentence should “correspond” to a dependency labelled, say, `obj` in the target sentence, we train a simple maximum likelihood estimate model of the conversion from source-side dependency labels to target-side dependency labels. To do this we look at the word-alignments generated in the course of phrase-table extraction, and parse trees on both sides of the training corpus. For each source side dependency label we thus construct a probability distribution over target side dependency labels, according to which a source dependency label is “normalised” to a target dependency label before consulting our model of target side linearisation offsets.

Note that the parsed parallel corpus used to train this model need not be the same parallel corpus as is used for phrase-table extraction (and in general, a much smaller one will suffice). The only requirements are that the source-side dependency parses are comparable to the parses of the input sentences that will be used at translation time, and that the target-side dependency parses are comparable to those on which the dependency-linearisation model is trained.

3.4.3. *Lexicalisation of the distortion model*

In order to allow for idiosyncratic linearisation properties of particular lexical items, we permit conditioning of distortion model probabilities not only on the label of the dependency being linearisation but also on the target-side lexical items appearing at each end of the dependency. If we have not observed enough instances of the (label, *gov_word*, *dep_word*) triple in the training data, we back off to less fine-grained conditions, the final option being to condition only on the (normalised) dependency label itself (as presented in the body of this paper).

3.5. DISCUSSION

This method allows only a subset of the foreign phrases which standard phrase-based systems allow. Such systems require that the words in a phrase be contiguous in the original sentence; our method requires this and more. For example, the phrase **THE SAW** in our example could be used by most phrase-based systems, but not by this system. It does not appear in the tree in Figure 5.

We can identify two justifications for not considering this phrase. First, it should be advantageous to translate words together in a phrase if there are dependencies between them, so that the English translations can be “ready made” with any agreement morphology which may result from the dependency. There is no dependency between **THE** and **SAW**. Second, it is not clear what dependency would link the resulting phrase **THE SAW** to, for example, (the phrase containing) the word **WOMAN**; **THE** is its dependent by a dependency of type *det*, while **SAW** is its governor by a dependency of type *obj*. If orderings are to be scored based on the kind of model described above, the distortion score for a hypothesis translating **THE SAW** as a phrase would not be defined.

Note that a phrase such as *he said* is *not* excluded by this technique, because while these two words do not typically form a syntactic constituent, they are linked by a dependency. Attempts to improve on phrase-based systems’ choices of phrases on the basis of *constituent* trees will penalise or reject such subject-verb combinations. Dependency trees are more forgiving, since the conversion from constituency

h_0	h_1	h_2
$\{\}$ 0 $\{\}$ ϵ	$\{1, 2\}$ 1 $\{((\text{subj}, 6, 1), 1)\}$ <i>he</i>	$\{1, 2, 3, 4, 5, 6\}$ 5 $\{\}$ <i>he saw the tall woman</i>

Figure 6. A sequence of three hypotheses searched during decoding

trees to dependency trees abstracts away from hierarchy among elements that share a governor; we can think of a dependency tree as representing an equivalence class of constituency trees. Fox (2002) has found that dependency structures are generally more faithfully maintained by human translations than constituency structures are.

4. Decoding

4.1. THE SEARCH SPACE

Three search states, or hypotheses, are shown in Figure 6. The start state or initial hypothesis is h_0 . The first line in the boxed representation of h_0 indicates the set of (indices of) foreign words covered so far (currently empty). The second line shows the number of dependencies among the translated phrases (currently zero) and any “half-translated” dependencies (currently none). The third line shows the translation so far, built up strictly left-to-right.

Suppose we are translating the foreign sentence whose possible compressions are illustrated in Figure 5. To produce a new hypothesis from h_0 , we must choose a phrase from Figure 5 and an English translation for that phrase. Suppose we choose the phrase **MAN THE** and the English translation *he*. Then the resulting hypothesis h_1 covers words 1 and 2 from the input, which are linked by one dependency (the *det* dependency); and the *subj* dependency with governor word 6 and dependent word 1 would be “half-translated” (this will be discussed further shortly). The cost of this step through the hypothesis space is just the product of the phrase translation cost and the language model cost. The distortion model only plays a role when considering the relative ordering of phrases, so it has nothing to do yet.

$$P(h_1) = P(h_0) \times P_\phi(\mathbf{he} | \mathbf{MAN THE})^{\lambda_\phi} \times P_\ell(\mathbf{he} | h_0)^{\lambda_\ell} \quad (4)$$

From h_1 we might next choose the foreign phrase **WOMAN TALL THE SAW**, and the translation *saw the tall woman*. This phrase “contains” three dependencies (*obj*, *det* and *amod*), and also “completes” the *subj* dependency linking it to **MAN THE**, so the new hypothesis h_2 covers

five dependencies (the entire sentence). The cost of this step through the hypothesis space is the product of the phrase translation cost, the language model cost, and the cost of these two phrases being ordered as they are, given the `subj` dependency between them:

$$P(h_2) = P(h_1) \times P_\phi(\textit{saw the tall woman} | \textit{WOMAN TALL THE SAW})^{\lambda_\phi} \\ \times P_\ell(\textit{saw the tall woman} | h_1)^{\lambda_\ell} \times P_d(-1 | \textit{subj})^{\lambda_d}$$

Note that by selecting a particular set of phrases which cover the foreign sentence, we implicitly select a particular compression of the original dependency tree in the course of decoding. In this example, by selecting the phrases `MAN THE` and `WOMAN TALL THE SAW`, we have implicitly selected the compression in Figure 4(b).

The reason for recording the set of half-translated dependencies in the form shown above (i.e. a set of tuples like $((\textit{subj}, 6, 1), 1)$) is that it permits hypothesis recombination. The offset for any newly-completed dependencies (eg. the -1 for the `subj` dependency above) could be computed, without storing this information in the hypothesis structure, simply by looking back at the partial English translation to see “how far back” the other end of the dependency occurs. But the set of half-translated dependencies as presented above contains all and only the information required to determine all future distortion model scores. If two hypotheses have the same half-translated dependency set (as well as the same set of words covered), then they can be combined.

4.2. FUTURE COST ESTIMATION

For the purposes of future cost estimation we assume that any dependency which has not yet been linearised (whether it is “half-translated” or completely untouched as yet) will be linearised with the best possible offset which is still possible. In the case of completely untouched dependencies (dependencies of which neither the governor nor the dependent have yet been linearised), all offsets are possible so we take the “unrestricted maximum” score. A half-translated dependency, however, will have a restricted range of offsets that are still possible for it.

For example, consider hypothesis h indicated below. This would be the penultimate step in a derivation of a full (although rather terrible) translation of the sentence considered in subsection 4.1. The subject and object subtrees have been translated, and only the verb remains.

$$h : \begin{array}{l} \{1, 2, 3, 4, 5\} \\ 3 \quad \{((\textit{obj}, 6, 3), 3), ((\textit{subj}, 6, 1), 1)\} \\ \textit{tall lady the the guy} \end{array}$$

The eventual offset for the linearisation of the `obj` dependency from this point can not be any greater than -3 , and similarly the eventual offset for the linearisation of the `subj` dependency can not be any greater than -1 . These are the offsets we would use if the other half of these dependencies were translated straight away, and the only way they could in principle change (if the sentence were longer than our extended example) is if we add more words to the English translation before finally translating these other halves. The future cost estimation for the distortion model for h is therefore

$$\left(\max_{n \leq -3} P_d(n|\text{obj})\right)^{\lambda_d} \left(\max_{n \leq -1} P_d(n|\text{subj})\right)^{\lambda_d}$$

4.3. PRUNING

Standard phrase-based systems avoid a bias against “more complete” hypotheses by arranging hypotheses in stacks, where each stack contains all the hypotheses which have covered a certain number of foreign words. In this new system however, two hypotheses with the same number of foreign words covered might not be comparable, because they have completed a different number of dependencies (and thus will include a different number of reordering probabilities). A stack must therefore be characterised by a number of foreign words covered and a number of dependencies covered. This is the reason for storing the number of completed dependencies in each hypothesis.

Also, in the same way that standard phrase-based systems make hypotheses which have covered the same number of words compete, irrespective of how many phrases they have used to do it, we make hypotheses which have covered the same number of dependencies compete, no matter how many phrases they have used to do it. (This provides a slight bias towards longer phrases.) This is the reason for adding any dependencies which are “contained in” the phrase being translated (eg. the `det` dependency when we translate [MAN THE]), and not just the dependencies *between* translated phrases.

5. Experiments, Results and Discussion

5.1. EXPERIMENTS AND RESULTS

We conducted Czech-English translation experiments, utilising the Prague Czech-English Dependency Treebank (PCEDT).³ We trained on the aligned sentence-pairs in the training section of the PCEDT’s PTB

Table II. Results for Czech-English translation experiments

	Available spans	Distortion models			BLEU	EDPM
		Pharaoh	Dep.	Lex. Dep.		
(i)	Pharaoh spans	✓			33.28	56.29
(ii)	Random spans	✓			20.53	45.86
(iii)	Licensed spans	✓			32.54	55.35
(iv)			✓		24.13	49.65
(v)		✓	✓		31.91	54.86
(vi)		✓		✓	31.82	55.19

component plus the aligned pairs in the Reader’s Digest component (65,110 sentences). We used the development and test sections (256 sentences each) of the PTB component. Development corpora were used to tune model weights via minimum error rate training (Och, 2003).

Table II summarises results. Our evaluation used both BLEU and the dependency-based EDPM metric (Kahn et al., 2009), but the overall pattern of results does not differ across these two metrics. The complete system described in this paper (line vi) underperforms the baseline (line i). Lines (ii) through (v) summarise additional experimentation seeking insight into the system’s (lack of) performance. We varied the spans of the input sentence which we allow ourselves to cover by a single phrase (“Available spans”), and the model(s) used to assign a score to a particular ordering of phrases (“Distortion models”).

With respect to the possible spans, “Pharaoh spans” refers to the entire set of spans that would be considered by a standard phrase-based decoder, namely, all chunks of the input sentence up to a certain size (specifically, seven words). “Licensed spans” refers to the subset of spans licensed by the input sentence’s dependency parse, as described in section 3.2. Holding the distortion model constant, comparing lines (i) and (iii) shows that restricting the model to only licensed spans resulted in only a small cost, despite a 54% average reduction in the size of the set of available spans. This suggests that dependency-based licensing for phrases may be on the right track. Line (ii) reinforces this observation: sampling a *random subset* of the spans used in line (i), equal in size to the set from line (iii), drops performance hugely compared to the principled dependency-based licensing.⁴ Some further analysis reveals a striking reason for this: although only 46% of the spans available in line (i) are available in line (iii), over 99% of the *phrases* available in line (i) are still available in line (iii). In other words, the spans of the input sentence which violate our dependency restriction were almost

never candidates for translation as a phrase in line (i) anyway, since no entry in the phrase table matched them. This indicates that the phrase-extraction process was already conforming to the idea behind our phrase-licensing scheme, perhaps as a result of the morphological richness of Czech. In other languages where the dependency structure is not as clearly encoded in the surface string, the phrase-extraction process may leave more of this work to be done at decoding time.

Of the three distortion models: “Pharaoh” refers to the standard monotonic-biased distortion model inherited from speech recognition, “Dep.” refers to the dependency-based model described in section 3.2 *without* lexicalisation (i.e. offsets are conditioned solely on dependency label), and “Lex. Dep.” refers to the dependency-based model *with* lexicalisation (i.e. offsets are conditioned on dependency label and the dependent word, as per subsection 3.4.3).⁵ Lines (iv), (v), and (vi) summarise results using these distortion model variants. The conventional distortion model outperforms the dependency-based variants.

We conducted additional experimentation using Arabic-English as the language pair, but unfortunately the results also failed to provide improvements over the baseline. We connect this finding to the observation that phrase-based models for this language pair do perfectly well in comparison to hierarchical phrase-based models, and the fact that various properties of Arabic (e.g. long run-on sentences, ubiquitous use of NP NP-modification *idafa* constructions) may make Arabic more difficult to parse correctly than Czech.⁶

5.2. DISCUSSION

This work was motivated by the idea from Hwa et al. (2002) and Fox (2002) that translations often preserve dependency structure. However a tempting suspicion, given the lack of improvement that this system has brought, is that the automatically-produced dependency parses on which it relies are not as accurate a representation of the commonalities between source sentence and reference translation as intuition might lead us to believe — either because this is a fundamental fact about dependency parses (apparently *contra* Hwa et al. (2002)), or because automatic parsers are not yet sufficiently accurate for this application. Some suggestive evidence for this can be found in relevant literature: for example, the reordering model from Tromble and Eisner (2009) does not improve translation quality when its features are based on dependency parses, but does yield improvements with features sensitive to surface-oriented, non-hierarchical properties; and Galley and Manning (2008) present a system very similar to ours that arranges flat phrases in a hierarchical structure, which does yield an improvement

over the phrase-based baseline, but these hierarchical structures are formally syntax-based rather than linguistically syntax-based (i.e. they are not constrained by an independent parser). Linguistically syntax-based systems that *have* yielded improvements over baselines (Quirk et al., 2005; Shen et al., 2008) use translation units that are essentially subtrees extracted from a training dependency corpus; this suggests that the more abstract “projection” relationship that we have assumed between source-language dependency structures and flat target-language translation units may therefore be problematic. The Czech dependency parses we used do often, upon casual inspection, differ in dependency structure from the reference translations.

The model we have presented was designed to tolerate some of the obvious problems that an MT system can run into when constrained by a parse tree: first, by using dependency trees rather than constituency trees (cf. section 3.5), and second, by permitting parts of trees to be “flattened” so that any erroneous structure may be disregarded. But to the extent that enforcing compliance with dependency parses remains an issue, it may be useful to somehow incorporate the ideas we present here as “soft constraints” (Marton and Resnik, 2008; Chiang et al., 2008). We have not found a way to do this when only a single parse tree of the source sentence is provided, but a more feasible variation may be to protect the system from suboptimal parses by providing an n -best list of parses (or a packed representation thereof (Dyer and Resnik, 2010)). A simple version of this would simply begin the current decoding process with n distinct empty hypotheses (one for each tree); a more useful approach would allow hypotheses to only gradually commit themselves to particular trees, starting from an initial state that is compatible with every parse tree and only cutting themselves off from a particular tree when they make a choice that is incompatible with it, but this would require more significant modifications to the decoder.

Some other, smaller-scale modifications to the system presented here also suggest themselves. First, we can imagine conditioning offset probabilities on the size of a dependent subtree, in an attempt to permit adjustments for “heaviness”, i.e. large constituents can appear in a non-canonical extraposed positions (Shen et al., 2009). Ideally we would measure the heaviness of a constituent in target-side words, but to permit scoring of partial translations we would need to approximate this (presumably quite reasonably) using the number of source-side words. Second, punctuation tokens are treated no differently from words in the PCEDT corpora we used, so they appear in dependency trees. We suspect it would be more useful to treat the reordering of words/phrases separately from the positioning of punctuation: one could remove all punctuation from training corpora and input trees, apply the model

as described in this paper to these punctuation-free sentences, and then use an independently-trained punctuation prediction model to add punctuation to the output (Lee et al., 2006).

6. Conclusion

In this paper, we explored a richer, linguistically informed reordering model while otherwise remaining strictly within the confines of phrase-based translation. The hope was to introduce an approach that could take advantage of monolingual syntactic dependencies, while still retaining a conventional statistical MT framework involving contiguous word strings as translation units, with the attendant advantages of conceptual simplicity, ease of implementation, availability of existing tools for training and tuning, and the ability to integrate seamlessly with other log-linear modeling features.

From a technical standpoint, we found it interesting to explore the idea that portions of a source syntactic dependency tree can be “compressed” into the translation units licensed by standard flat-phrase extraction. Despite the negative results of the distortion model we have introduced, restricting the search space to only translations using dependency-licensed phrases — a significant reduction in the range of possible spans to be covered — had only a very small negative effect. This suggests that there is something fundamentally sound about the intuition that linearisation of flat phrases can be thought of in terms of cross-phrase head-modifier relationships.

Notes

¹ A Python implementation of the decoder, training scripts, and data are available at <http://www.ling.umd.edu/~timh/decoder/decoder-distrib.tgz>.

² Tromble and Eisner (2009) present a recent variant of this pre-preprocessing approach, which differs in that it learns a model of reorderings purely on the basis of an automatically-aligned bitext, without using any parse trees.

³ See Čmejrek et al. (2004) and <http://ufal.mff.cuni.cz/pcedt/>. We used the provided parse trees from the Collins parser.

⁴ Also, lines (ii) and (iii) use phrases of maximum length 4 whereas the baseline uses 7. Permitting phrases of length greater than 4 produces an unnecessary explosion in the number of possible compressions of the dependency tree; line (ii) also uses 4 as a limit to provide a clear comparison. However, less than 0.01% of the phrases usable on our test set by the baseline system were of length longer than 4.

⁵ Note that the dependency-based models are only compatible with the “Licensed phrases” option, since they cannot assign scores to orderings of arbitrary phrases.

⁶ Chris Dyer and Chris Manning, personal communication.

References

- Brown, P. F., J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin: 1990, ‘A statistical approach to machine translation’. *Computational Linguistics* **16**(2), 79–85.
- Chiang, D.: 2005, ‘A hierarchical phrase-based model for statistical machine translation’. In: *Proceedings of ACL*. pp. 263–270.
- Chiang, D., Y. Marton, and P. Resnik: 2008, ‘Online Large-Margin Training of Syntactic and Structural Translation Features’. In: *Proceedings of EMNLP*. pp. 224–233.
- Čmejrek, M., J. Cuřín, and J. Havelka: 2004, ‘Prague Czech-English Dependency Treebank: Any Hopes for a Common Annotation Scheme?’. In: *Proceedings of HLT/NAACL 2004 Workshop: Frontiers in Corpus Annotation*. pp. 47–54.
- Collins, M., P. Koehn, and I. Kučerová: 2005, ‘Clause restructuring for statistical machine translation’. In: *Proceedings of ACL*. pp. 531–540.
- Dyer, C. and P. Resnik: 2010, ‘Forest translation’. In: *Proceedings of NAACL-HLT*.
- Fox, H. J.: 2002, ‘Phrasal Cohesion and Statistical Machine Translation’. In: *Proceedings of EMNLP*. pp. 304–311.
- Galley, M., M. Hopkins, K. Knight, and D. Marcu: 2004, ‘What’s in a translation rule?’. In: *Proceedings of HLT-NAACL*. pp. 273–280.
- Galley, M. and C. D. Manning: 2008, ‘A Simple and Effective Hierarchical Phrase Reordering Model’. In: *Proceedings of EMNLP*. pp. 848–856.
- Gildea, D.: 2003, ‘Loosely tree-based alignment for machine translation’. In: *Proceedings of ACL*. pp. 80–87.
- Hunter, T. and P. Resnik: 2009, ‘Extending Phrase-Based Decoding with a Dependency-Based Reordering Model’. Technical Report UMIACS-TR-2009-15, LAMP-TR-152. Available at <http://hdl.handle.net/1903/9782>.
- Hwa, R., P. Resnik, A. Weinberg, and O. Kolak: 2002, ‘Evaluating Translation Correspondence using Annotation Projection’. In: *Proceedings of ACL*. pp. 392–399.
- Kahn, J. G., M. Snover, and M. Ostendorf: 2009, ‘Expected Dependency Pair Match: Predicting translation quality with expected syntactic structure’. *Machine Translation*. Published online 31 Oct. 2009.
- Koehn, P., F. J. Och, and D. Marcu: 2003, ‘Statistical phrase based translation’. In: *Proceedings of HLT-NAACL*. pp. 127–133.
- Lee, Y.-S., S. Roukos, Y. Al-Onaizan, and K. Papieni: 2006, ‘IBM Spoken Language Translation System’. In: *Proceedings of TC-STAR Workshop*. pp. 13–18.
- Marton, Y. and P. Resnik: 2008, ‘Soft Syntactic Constraints for Hierarchical Phrase-Based Translation’. In: *Proceedings of ACL*. pp. 1003–1011.
- Och, F.: 2003, ‘Minimum Error Rate Training for Statistical Machine Translation’. In: *Proceedings of ACL*. pp. 160–167.
- Och, F. J., C. Tillman, and H. Ney: 1999, ‘Improved Alignment Models for Statistical Machine Translation’. In: *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora*. pp. 20–28.
- Quirk, C., A. Menezes, and C. Cherry: 2005, ‘Dependency Tree Translation: Syntactically Informed Phrasal SMT’. In: *Proceedings of ACL*. pp. 271–279.
- Shen, L., J. Xu, and R. Weischedel: 2008, ‘A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model’. In: *Proceedings of ACL*. pp. 577–585.

- Shen, L., J. Xu, B. Zhang, and S. M. R. Weischedel: 2009, ‘Effective Use of Linguistic and Contextual Information for Statistical Machine Translation’. In: *Proceedings of EMNLP*. pp. 72–80.
- Tromble, R. and J. Eisner: 2009, ‘Learning Linear Ordering Problems for Better Translation’. In: *Proceedings of EMNLP*. pp. 1007–1016.
- Wu, D. and H. Wong: 1998, ‘Machine Translation with a Stochastic Grammatical Channel’. In: *Proceedings of ACL-COLING*. pp. 1408–1415.
- Xia, F. and M. McCord: 2004, ‘Improving a statistical MT system with automatically learned rewrite patterns’. In: *Proceedings of COLING*. pp. 508–514.
- Yamada, K. and K. Knight: 2001, ‘A syntax-based statistical translation model’. In: *Proceedings of ACL*. pp. 523–530.