Shallow-Syntax Phrase-Based Translation: Joint versus Factored String-to-Chunk Models

Mauro Cettolo, Marcello Federico, Daniele Pighin and Nicola Bertoldi
Fondazione Bruno Kessler
via Sommarive, 18 - I-38100 Povo di Trento, Italy
<*surname*>@fbk.eu

Abstract

This work extends phrase-based statistical MT (SMT) with shallow syntax dependencies. Two string-to-chunks translation models are proposed: a factored model, which augments phrase-based SMT with layered dependencies, and a joint model, that extends the phrase translation table with *microtags*, i.e. per-word projections of chunk labels. Both rely on n-gram models of target sequences with different granularity: single words, microtags, chunks. In particular, n-grams defined over syntactic chunks should model syntactic constraints coping with word-group movements. Experimental analysis and evaluation conducted on two popular Chinese-English tasks suggest that the shallow-syntax joint-translation model has potential to outperform state-of-the-art phrase-based translation, with a reasonable computational overhead.

1 Introduction

Many promising efforts in MT are nowadays toward the effective and efficient integration of syntactic knowledge into the statistical approach. As a matter of fact, state-of-the-art phrase-based translation (Koehn et al., 2003) seems to face severe limitations when applied to language pairs, like Chinese-English, that significantly differ in word order and syntactic structure. In principle, phrase-based statistical MT (SMT) can permit rather long word movements; in practice, translation hypotheses computed during search are scored by word-based n-gram language models (LMs) which capture only rather local dependencies.

Syntax-driven models were proposed to overcome limitations of phrase-based approaches regarding word-reordering and structural coherence of translations. While standard phrase-based systems typically rely on n-gram models defined over linear structures (sequences), syntax-based SMT exploits stochastic dependencies defined over tree structures. Figures 1.a and 1.d graphically show the dependencies in these two models.

Recently, *factored translation models* were proposed in order to augment phrase-based SMT with layered dependencies. The original idea was to reduce data-sparseness by factoring the surface representation of words into base-form, morphology, and part-of-speech (Koehn and Hoang, 2007).

The present work extends phrase-based SMT with shallow syntax dependencies at both word and chunk levels. In particular, syntactic constraints coping with word-group movements are modeled by an n-gram model defined over syntactic chunks rather than single words. Moreover, two alternative *string-to-chunks* translation models are discussed: a factored model, defined along the line of (Koehn and Hoang, 2007), and a *joint* model, that extends the phrase translation table with *microtags* (as we call the per-word projections of chunk labels, see Section 3.1) on the target language side. Both models rely on n-gram models of target sequences with different granularity: single words, microtags, chunks.

Figures 1.b and 1.c depict the dependencies involved in the two models. In our factored model, the chunk layer is built in a deterministic way above standard factors whose top-most layer is that of microtags. In the joint model, words and microtags are

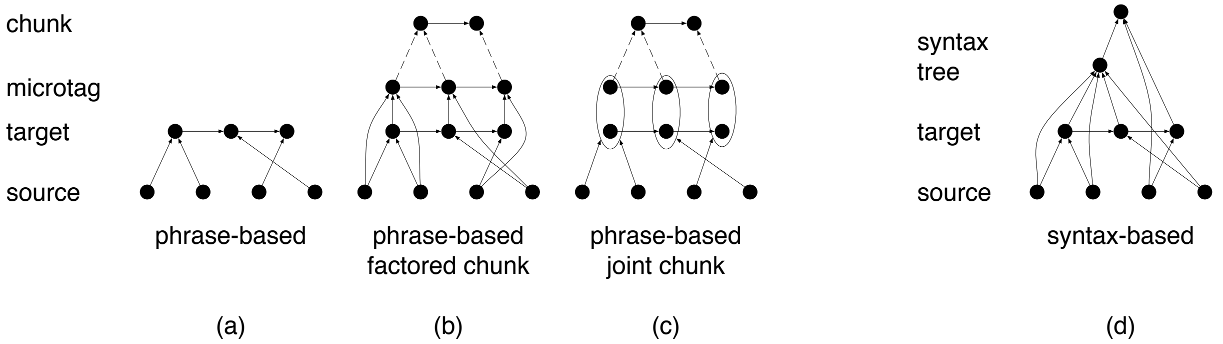

Figure 1: Stochastic dependencies used by different translation models. Source phrases are translated into: (a) target phrases in phrase-based translation; (b) target phrases and microtag sequences in the factored model; (c) pairs of phrases and microtag sequences in the joint model; (d) nodes of a full syntactic parse in the syntax based model.

tightly tied to form a single layer, above which the chunk layer is built as in the factored model.

Our models were implemented under the Moses (Koehn et al., 2007) platform[1], a popular open source toolkit. In order to compare the two string-to-chunks translation models, both in terms of computational efficiency and translation accuracy, we ran experiments on two Chinese-English translation tasks: traveling domain expressions, as proposed by the IWSLT workshop, and news translation, as prepared by the NIST MT workshops. Due to its limited size, the former dataset was used to analyze from the computational cost point of view the models under investigation. Conversely, evaluations were performed on the NIST task, which consists of syntactically rich sentences whose translation can more clearly benefit from the introduction of chunk-level dependencies and constraints.

2 Previous Work

Recent literature reports on several approaches for integrating syntactic knowledge into SMT. As a simple classification criterion, we consider the point at which syntactic information is exploited within the typical processing chain of SMT: pre-processing, decoding, and rescoring.

Several papers discussed the use of syntactic reordering rules to pre-process the input string so that it matches better the structure of target language (English). Examples of considered source languages are German (Collins et al., 2005), Chinese (Wang et al., 2007) and Arabic. The approaches discussed in those papers permit relevant re-ordering phenomena at the syntactic level to address; nevertheless, to our view they suffer severe limitations: they require human skills specific to each language pair and their impact is in general limited to a small number of rules. Examples of automatic reordering of source strings are presented in (Zhang et al., 2007) and (Habash, 2007) for the Chinese and Arabic languages, respectively.

Concerning the application of syntactic information to re-score N-best lists of translations from Chinese to English, a spectrum of techniques was investigated (Och et al., 2004). These range from shallow syntactic features, namely a part-of-speech (POS) LM defined over POS projected from the source language to the target language, to parse tree probabilities. An alternative approach was proposed in (Chen et al., 2006), where re-ordering rules at the level of single POS or POS-phrases are learned from the aligned training data. Similarly to (Och et al., 2004), POS information is computed on the source language. Both approaches showed some improvement over a standard baseline, but their scope and consequently impact is clearly limited, given that N-best lists represent a small fraction of the actual search space explored by the search algorithm.

To overcome this limitation, the only way is to directly integrate syntactic knowledge in the search algorithm. Prominent examples in the literature are:

- Hierarchical model (Chiang, 2005), in which context free rules are inferred from aligned

[1] Available from http://www.statmt.org/moses/

string-to-string pairs (notice: no parsing is required).

- Syntax model (Galley et al., 2006), in which syntactic translation rules are inferred from aligned tree-string pairs and parse trees are computed on the target language.
- Dependency tree-lets (Quirk et al., 2005), in which a dependency tree-based reordering model is inferred from aligned string-tree pairs. Parsing is performed on the source language and a corresponding dependency grammar is inferred on the aligned target side.

The above approaches showed in several occasions to outperform phrase-based SMT in terms of translation quality. Unfortunately, the corresponding search procedures are more complex and difficult to implement than those for phrase-based SMT.

Recently, (Hassan et al., 2007) introduced syntactic constraints into phrase-based SMT by 'syntactifing' target language phrases with supertags. In order to account for the grammaticality of translation hypotheses, the supertags LM score is weighted with respect to the number of compositional constraints violated by the n-gram sequences.

Supertags extracted from parse-trees were also investigated in (Birch et al., 2007) for embedding syntactic knowledge into factored models. These works showed that tree-based structural dependencies can also be embedded into a phrase-based decoder. Our work goes along this direction by introducing three main novelties:

- we assume that word-reordering just requires proper construction at the chunk and word levels;
- n-gram models are also defined over chunks: in this way, longer word spans are effectively covered;
- we propose a joint model that simplifies significantly the factored model.

3 Shallow Syntax Models

Our models integrate the word level of the target language with shallow-syntactic data obtained with an automatic chunker. The goal is to obtain better-formed translations by aiding phrase selection and reordering with constraints enforced at the syntactic level. The kind of information that we encode is described in Section 3.1.

A way to encode non-lexical information in a SMT model is to use *factored* translation models (Koehn and Hoang, 2007): the translation unit is no more a (string of) word(s) but a vector of factors; each factor represents a different level of annotation that can enrich the surface form with grammatical knowledge, such as lemma, part-of-speech, morphological features and so on.

An alternative solution, which we refer to as a *joint* model, consists in using as target tokens the concatenations of the symbols from the different layers.

As the comparison between the joint and the factored model is central to this work, they will be further discussed in Sections 3.2 and 3.3. Section 3.4 compares complexity aspects of the two approaches.

3.1 Using chunks to support SMT

The information that we encode in the syntactic layer is derived from the shallow parses of the target sentences. Each word w in a chunk labeled TAG is assigned a *microtag*:

- $TAG($ if w is the first word in X;
- $TAG)$ if w is the last word in X;
- $TAG+$ if w is internal to X;
- TAG if the chunk consists of just one word.

Microtags preserve the information about the chunk and allow us to reconstruct the sequence of chunk labels based on the microtag sequence, e.g. the microtags *VP NP(NP) PP(PP)* correspond to the chunk sequence *VP NP PP*. An example of micro and chunk labeling of a sentence is shown in Figure 2.b.

The microtag model is a standard n-gram model which captures the internal structure of chunks and patterns across chunks. It should be able to enforce constraints in the search space that would prevent incompatible phrases to be adjacent in the translation, e.g. if the last translated symbol is an *NC(* or *NC+* we would like to restrict the search to microtag phrases beginning with *NC+* or *NC)* (intra-chunk consistency).

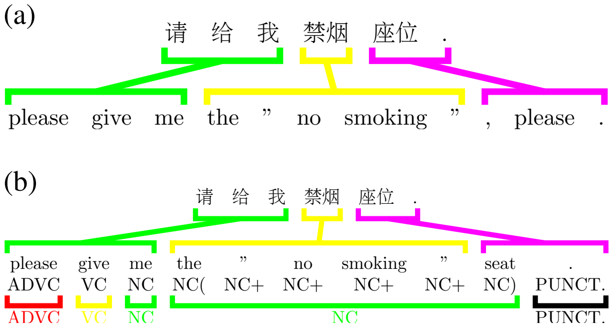
(a)
请 给 我 禁烟 座位 .
please give me the ” no smoking ” , please .
(b)
请 给 我 禁烟 座位 .
please give me the ” no smoking ” seat .
ADVC VC NC NC(NC+ NC+ NC+ NC+ NC) PUNCT.
ADVC VC NC NC PUNCT.

Figure 2: (a) Example of translation by a standard phrase-based SMT system. (b) The same sentence translated by our shallow-syntax aided SMT system. (One of the references is “please reserve a non-smoking seat .”)

Also the model of sequences of chunks is a standard n-gram model. Chunks can consist of more words: during decoding, the chunk model must be queried once for each chunk, i.e. in an asynchronous manner with respect to the other n-gram models. The chunk model is expected to filter out translations that exhibit unlikely syntactic structure, e.g. that do not include verbal chunks or that sport long sequences of verb chunks that do not interleave with typical predicate argument chunks, such as nominal or prepositional ones (inter-chunk consistency).

As an example of intra-chunk consistency, consider the alignment examples shown in Figures 2.a and 2.b automatically obtained for one of the Chinese-to-English tasks we worked on. The first results from a standard phrase-based SMT model (baseline), whereas the latter makes use of syntactic information. The word “seat”, which is missing in the baseline translation, allows to “close” the nominal chunk it belongs to in the chunk-aided translation. The resulting microtag sequence, corresponding to a locally well-formed syntactic interpretation of the lexical tokens sequence, is likely to be assigned a high probability by the corresponding n-gram model as it is quite common in the training data. Conversely, sequences in which *NC+* is not followed by *NC+* or *NC)* have never been observed and therefore tend to receive lower probability values.

Regarding inter-chunk consistency, consider again the example in Figure 2.b and look at the chunk sequence *VC NC NC*. This sequence is typical of double object verb forms, such as the predicate “give” in the example. In this case the nominal chunks are quite simple and a 6-gram model would be able to capture this dependency, but for more complex, longer chunks this kind of shallow predicate-argument relation couldn’t be handled by a traditional n-gram model. Conversely, our representation would be able to account for it as the chunk-level sequence would be just the same.

In the following sections, we detail the two string-to-chunks models. For the sake of simplicity, during the discussion we will refer to the single word as a translation unit; the generalization to phrase based MT is straightforward.

3.2 Factored String-to-Chunks Translation

In factored translation models (Koehn and Hoang, 2007) a vector of source factors is translated into a vector of target factors. For both languages, the first factor generally encodes the lexical level whereas the others could capture the most diverse information, from morphological features to semantic annotations. For each target factor involved, an appropriate n-gram model should be estimated.

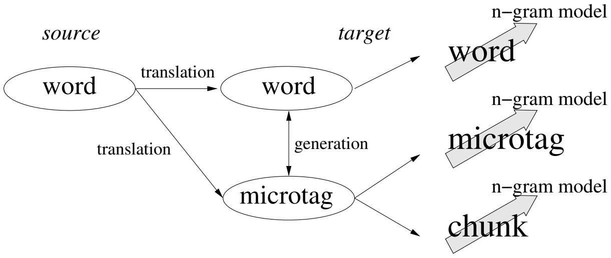

Figure 3: Illustration of the factored chunk model. The word and the microtag models are queried on a per-word basis. The chunk n-gram model is invoked whenever a chunk is closed. A generation step limits the number of *(word, microtag)* pairs.

Our factored model for chunk-based SMT employs just one source factor (the Chinese words) and two factors on the target side: the English words and their corresponding microtags. Each source word is translated both into a target word and into a microtag by two distinct translation steps. A generation step is performed to limit the *(word, microtag)* combinations to the pairs that are coherent with events observed in the training data. Figure 3 illustrates this arrangement.

The word and microtag n-gram models are queried every time a new word is added to a translation hypothesis. This is not true for the chunk model, whose granularity is coarser as generally chunks are not in one-to-one correspondence with words. Instead, for every explored sequence of microtags the corresponding sequence of chunks is built. The chunk model is queried only when a chunk is closed, so that the score is provided once for each chunk.

The microtag sequence in a translation hypothesis may be inconsistent. For example, a *VC(* may be followed by an *NC(* instead of the correct *VC+* or *VC)*. These situations are resolved by forcing the closure of the incomplete chunk. In this example, we would assume that the first *VC* chunk has been closed and a new *NC* chunk opened.

3.3 Joint String-to-Chunks Translation

The second solution relies on translation target units which are the concatenation of a target word and the corresponding microtag.

For both the word and the microtag level, a separate n-gram model is trained. Whenever a new *(word, microtag)* pair is to be added to a translation hypothesis the scores provided by the two models are combined. The behavior of the chunk model is just the same as described in the case of the factored model. Figure 4 illustrates the joint model for multi-layered SMT.

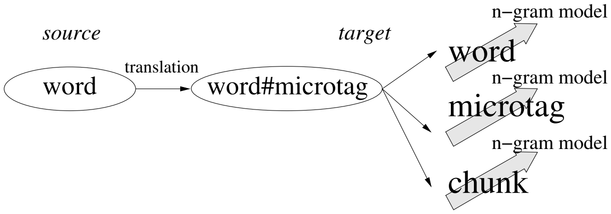

Figure 4: Illustration of the joint chunk model. Each Chinese word is mapped onto a *word#microtag* sequence. The chunk model is invoked asynchronously. There is no need for a generation step as all the possible pairs are those observed during training.

This joint approach does not require a generation step as the only possible *(word, microtag)* pairs are those observed at training time and that populate the translation tables.

3.4 Complexity of Models

For discussing this issue, let us refer to the Moses decoder, which implements an efficient decoding algorithm for SMT. It starts by generating the list of translation options, which are the possible translations of each input span given the models. The search space is built only on that list. In case of multiple factors, for a given span each phrase table (e.g. that of words and that of microtags) is queried to collect the list of possible translations. In theory, each element of a list should be paired with each element of other lists; in practice, this can be limited to events occurring in the generation table which links target factors according to what was observed in training data. Nevertheless, the number of translation options is typically much larger for multiple than for single factor models, like the standard phrase based SMT and our joint chunk model. Considering that the number of partial translations generated during decoding is an exponential function (limited by the beam search) of the number of translation options, we expect that multiple factors decoding is definitely more expensive than single factor one. A quantitative comparison between the two solutions will be carried out in the next section.

4 Evaluation

4.1 Translation Tasks

Experiments were carried out on a traveling domain, proposed by the 2007 IWSLT Workshop (Cettolo and Federico, 2007), and on a news domain proposed by the NIST 2006 MT Evaluation Workshop[2], from Chinese to English. Detailed figures about the employed training, development and test sets are reported in Table 1.

Translation performance are reported in terms of case-insensitive BLEU% and NIST scores. Statistical significance tests comparing performance of two systems were also applied. As proposed in (Koehn and Monz, 2006), a paired sign test on BLEU and NIST scores was performed on a 50-fold partition of the test set.

4.2 Data Annotation

The annotation of training data in terms of microtags is performed by the TreeTagger tool (Schmid,

[2] www.nist.gov/speech/tests/mt/

Task	Set	# of words	
		Source	Target
IWSLT	train	353K	377K
	dev 07	10.8K	12.3K
	test 07	3.2K	3.7K
NIST	train	83.1M	87.6M
	dev 02	23.7K	26.4K
	test 03	25.6K	28.5K
	test 04	51.0K	58.9K
	test 05	31.2K	34.6K

Table 1: Statistics of training, development and test sets. Development/test sets include multiple references: in table, average lenghts are provided.

1994). It is a part-of-speech tagger and chunker that employs *decision trees* to estimate transition probabilities. As a side effect of the tagging, contracted forms ('d, 'm, 's, etc.) and negations (not, n't) are separated from the preceding word, in order to be properly tagged.

4.3 Tuning

For experiments, we employed the Moses toolkit which includes tools to train the bilingual phrase tables and the distortion models given a word-aligned parallel corpus, and to optimize feature weights on a development set through a Minimum Error Rate training.

In particular, phrase-based translation models are estimated as follows. i) The training parallel corpus is word-aligned by means of the GIZA++ software tool (Och and Ney, 2003) in both source-to-target and target-to-source directions; ii) a list of phrase-pairs (up to 8 words) is extracted exploiting both word-alignments; iii) each phrase pair is associated with direct and inverse phrase-based and word-based probabilities.

This standard training procedure is straightforwardly applied to the baseline and the factored systems. Instead, for the joint system step ii) is anticipated by the concatenation of microtags to words; hence, target phrases in the joint model actually consist of *word#microtag* tokens rather than words.

Table 2 provides statistics on the phrase tables of the three models at study on the IWSLT task. In particular, the number of distinct source and target phrases, and the average number of translations per source phrase are given. Note that for the sake of a direct comparison of the chunk systems, we had to expand the two phrase tables and the generation table of the factored system into one equivalent phrase table comparable with that of the joint system. The expansion procedure simulates the way Moses generates the translation options. The larger number of the target phrases for the factored and joint models with respect to the baseline (+11% and +5%, respectively) suggests that the former models can be more affected by beam search pruning and, at least the joint model, by data sparseness.

system	# source phrases	# target phrases	Avg # trans
baseline	273K	277K	1.26
factored	”	307K	1.42
joint	”	291K	1.30

Table 2: Phrase table statistics for IWSLT task.

Concerning word reordering, the "orientation-bidirectional-fe" distortion model (Koehn et al., 2005) was estimated. Word-based 5-gram LMs are trained with modified Kneser-Ney discounting (Goodman and Chen, 1998), while micro and chunk 6-gram models with Witten-Bell discounting (Witten and Bell, 1991).

In decoding, for each model the parameters defining the beam have been set to values that limit the search errors as much as possible.

4.4 Experimental Results

We conducted a set of preliminary experiments and the analysis of proposed models on the IWSLT task. Thanks to its features, the IWSLT task offers a fast prototyping cycle, even for complex translation models, such as factored models.

Results of this investigation are reported in Table 3. Translation accuracy scores do not show clear nor statistically significant improvements over the baseline. However, they well compare with the official results of the evaluation campaign (Fordyce, 2007), taking into account that our models are trained on IWSLT training data only and that no rescoring stage was added to the standard decoding. Moreover, it must be noticed that sentences of the

IWSLT tasks are typically very short, with rather plain syntactic structure and many colloquial expressions. All these features limit very much the potential impact of syntax driven translation.

For allowing the comparison in terms of computational costs, the table provides the number of translation options (TrOpt) and the number of partial translations (GenTh) generated during decoding. These point out that the factored model is significantly more demanding than the joint model, both in terms of memory and time requirements. For this reason, we have so far been unable to set up an effective factored system on the NIST task, mostly due to overlong decoding time (whatever the size of LMs).

A more detailed discussion on computational issues of the considered approaches is provided in Section 5.

system	BLEU	NIST	TrOpt	GenTh
			$\times 10^3$	$\times 10^9$
baseline	35.4	6.28	155	1.08
factored	35.7	6.44	408	3.50
joint	35.1	6.33	193	1.61

Table 3: Results on the IWSLT task.

Experimental results on the NIST task are reported in Table 4 for the baseline and joint models only. The joint model outperforms the baseline system on all test sets. Statistical significance levels of the BLEU and NIST score differences range from α=0.06 to α=0.01. This evidence suggests two things: first, the potential of string-to-chunks models needs to be assessed on tasks where the syntactic structure of sentences is sufficiently complex; second, the joint model is an effective and very promising alternative to factored models towards the integration of shallow syntax dependencies into SMT.

Test	baseline/joint			
	BLEU	α	NIST	α
03	28.8 / 30.1	0.01	8.66 / 8.86	0.01
04	31.4 / 31.9	0.04	9.31 / 9.41	0.01
05	27.7 / 28.6	0.06	8.44 / 8.55	0.06

Table 4: Results on the NIST task with statistical significance levels.

Chi	Eng	system	microtags
冰箱	ice chest	factored	NC+ NC), NC(NC)
		joint	NC+ NC)
以上	a	factored	NC(, NC)
		joint	NC(

Table 5: Shallow syntax interpretations (microtags sequence) of phrase pairs for the chunk systems.

5 Discussion

First considerations can be drawn by looking at the statistics about the phrase tables from which the decoder extracts the translation options, reported in Table 2. On average, the factored model has 13% more translation options than the baseline model, the joint model only 3%. This difference is due to the method for extracting phrase pairs from the aligned training corpus, which is less constrained for the former than for the latter. It is worth noting that the set of translation options generated through the joint model is a subset of those generated by the factored model.

As expected, the difference is larger for short source phrases than for longer ones, as shown in Figure 5 which plots the average number of translations for any length of the source phrase. For instance, for source phrases of length 1, the factored model has 44% more translation alternatives than the joint model (3.13 vs. 2.18).

On one side, the over-generation provided by the factored model with respect to the joint model is positive because it allows to create shallow syntax interpretations of a target string which are not contained in the training data. As shown in Table 5, the new microtag sequence *NC(NC)* for “ice chest” is correct. On the other hand, it can happen that some new interpretations are wrong: indeed, it is very unlikely that the article “a” can close a noun chunk.

As the decoder exploits all translation options of the source phrase pairs (if no beam search is applied), it is straightforward that the factored system potentially has a search space significantly larger than the joint one. Hence, we expect that the former system is significantly less efficient than the latter in terms of decoding time.

This a-priori consideration is confirmed by the run-time behavior. As reported in Table 3, the factored and joint decoders compute a larger amount of translation options than the baseline (+163% and +25%, respectively) and accordingly generate a larger amount of partial translation hypotheses (+224% and +49%, respectively). Furthermore, we can state that the joint decoder is more efficient than the factored one at least by a factor of 2.

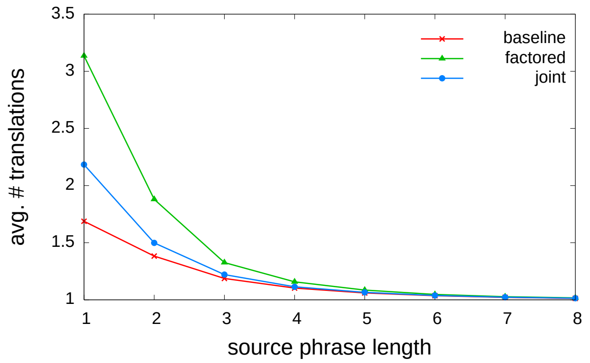

Figure 5: Average number of translation options per source phrase.

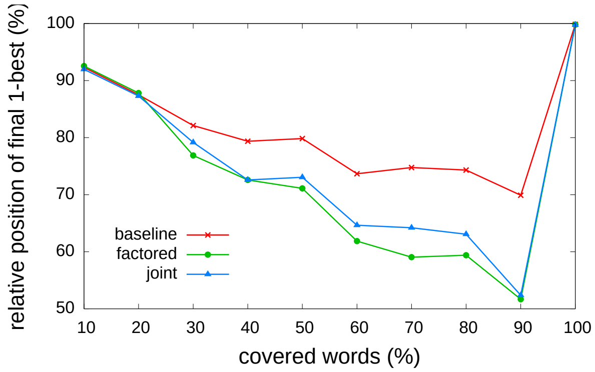

Figure 6: Relative position of the final 1-best during search with the three considered translation models.

Figure 6 provides a graphical hint on how the decoder explores the search space with the considered models. The three curves (one for each model) give the relative position of the final best hypothesis among the current translation hypotheses ranked by score. They are functions of the percentage of covered words and are computed by averaging over all the test sentences and scores of all partial hypotheses generated by the search algorithm. Generally speaking, the higher is the curve, the closer is the final 1-best to the current best, that is the less search errors are expected. It results that string-to-chunks models are more prone to search errors than the baseline model, that is for them the beam search has to be set with care. Since the joint model is significantly cheaper than the factored model in terms of complexity, as discussed above, it could be more easily deployed in large translation tasks involving training sets of billion of words.

6 Future Work

Our work on the introduction of chunk-level information in the SMT process is still in its early stages. The results on the large NIST dataset are encouraging and suggest that such information can indeed improve the translation accuracy. Unlike the factored model, the joint model seems to offer a good tradeoff between the potential accuracy improvement and the computational burden implied. Nevertheless, there are several research directions that might be explored in order to improve the benefits and reduce the drawbacks of string-to-chunks models.

More precise models could be obtained by introducing lexical dependencies in the microtag and chunk layers. In the case of microtags the lexicalization can be simply done on the lemma of the corresponding word, possibly taking into account statistical or linguistic hints. In the case of chunks the lexicalization involves the selection of a representative word among those that define the chunk; a possible choice could be the *chunk head*, that should be determined at search time.

A more fine-grained representation of the microtag layer could also be obtained by adding the size or structure of the chunk they come from. Several strategies may be compared in order to find an optimal compromise between the sparsity of the resulting n-gram model and its impact on the translation accuracy.

Other important issues involve the decoding algorithm. As stated, the chunk model is queried whenever a chunk is closed, that is in an asynchronous way with respect to the decoding steps, that are made on a target-word basis. As a consequence, partial theories covering the same source positions could be scored by a different number of models just because they are chunked in a different manner. The use of a chunk penalty should be investigated, similar to word and phrase penalties typically exploited, just to make translation hypotheses of different chunk length more comparable.

Finally, as suggested by Figure 6, dynamic pruning strategies could be applied during search in order to further reduce the run-time cost of string-to-chunks models: in fact, it seems that no additional search errors would occur if the search starts with a reduced beam which is enlarged step by step.

References

A. Birch, M. Osborne, and P. Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic.

M. Cettolo and M. Federico, editors. 2007. *International Workshop on Spoken Language Translation (IWSLT 2007)*. FBK-irst Trento, Italy.

B. Chen, M. Cettolo, and M. Federico. 2006. Reordering Rules for Phrase-based Statistical Machine Translation. In *Proc. of IWSLT*, Kyoto, Japan.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL*, pages 531–540, Ann Arbor, Michigan.

C. Fordyce. 2007. Overview of the IWSLT 2007 Evaluation Campaign. In *Proc. of IWSLT*, pages 1–12, Trento, Italy.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc of ACL*, pages 961–968, Sydney, Australia.

J. Goodman and S. Chen. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, August.

N. Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proc. of MT-Summit*, Copenhagen, Denmark.

H. Hassan, K. Sima'an, and A. Way. 2007. Supertagged phrase-based statistical machine translation. In *Proc. of ACL*, pages 288–295, Prague, Czech Republic. Association for Computational Linguistics.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proc of EMNLP-CoNLL*, pages 868–876.

P. Koehn and C. Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proc. of the Workshop on Statistical Machine Translation*, pages 102–121, New York City, NY, June.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT/NAACL*, pages 127–133, Edmonton, Canada.

P. Koehn, A. Axelrod, A. Birch Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proc. of IWSLT*, Pittsburgh, PA.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

F.J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

F.J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, et al. 2004. A smorgasbord of features for statistical machine translation. In *Proc. of HLT-NAACL*, pages 161–168.

C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. of ACL*, pages 271–279, Ann Arbor, Michigan.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. of the Int. Conf. on New Methods in Language Processing*, Manchester, UK.

C. Wang, M. Collins, and P. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proc.of EMNLP-CoNLL*, pages 737–745.

I.H. Witten and T.C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Inform. Theory*, IT-37(4):1085–1094.

Y. Zhang, R. Zens, and H. Ney. 2007. Improved chunk-level reordering for statistical machine translation. In *Proc. of IWSLT*, Trento, Italy.