Learning to Prune: Context-Sensitive Pruning for Syntactic MT

Wenduan Xu
Computer Laboratory
University of Cambridge
wenduan.xu@cl.cam.ac.uk

Yue Zhang
Singapore University of
Technology and Design
yue_zhang@sutd.edu.sg

Philip Williams and **Philipp Koehn**
School of Informatics
University of Edinburgh
p.j.williams-2@sms.ed.ac.uk
pkoehn@inf.ed.ac.uk

Abstract

We present a context-sensitive chart pruning method for CKY-style MT decoding. Source phrases that are unlikely to have aligned target constituents are identified using sequence labellers learned from the parallel corpus, and speed-up is obtained by pruning corresponding chart cells. The proposed method is easy to implement, orthogonal to cube pruning and additive to its pruning power. On a full-scale English-to-German experiment with a string-to-tree model, we obtain a speed-up of more than 60% over a strong baseline, with no loss in BLEU.

1 Introduction

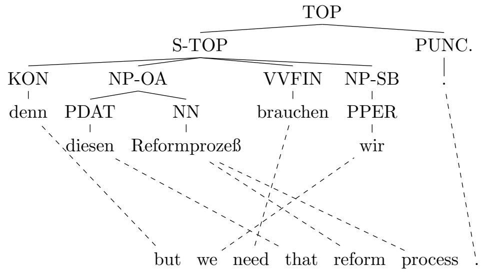
Syntactic MT models suffer from decoding efficiency bottlenecks introduced by online n-gram language model integration and high grammar complexity. Various efforts have been devoted to improving decoding efficiency, including hypergraph rescoring (Heafield et al., 2013; Huang and Chiang, 2007), coarse-to-fine processing (Petrov et al., 2008; Zhang and Gildea, 2008) and grammar transformations (Zhang et al., 2006). For more expressive, linguistically-motivated syntactic MT models (Galley et al., 2004; Galley et al., 2006), the grammar complexity has grown considerably over hierarchical phrase-based models (Chiang, 2007), and decoding still suffers from efficiency issues (DeNero et al., 2009).

In this paper, we study a chart pruning method for CKY-style MT decoding that is orthogonal to cube pruning (Chiang, 2007) and additive to its pruning power. The main intuition of our method is to find those source phrases (i.e. any sequence of consecutive words) that are unlikely to have any consistently aligned target counterparts according to the source context and grammar constraints. We show that by using highly-efficient sequence labelling models learned from the bitext used for translation model training, such phrases can be effectively identified prior to MT decoding, and corresponding chart cells can be excluded for decoding without affecting translation quality.

We call our method *context-sensitive pruning* (CSP); it can be viewed as a bilingual adaptation of similar methods in monolingual parsing (Roark and Hollingshead, 2008; Zhang et al., 2010) which improve parsing efficiency by "closing" chart cells using binary classifiers. Our contribution is that we demonstrate such methods can be applied to synchronous-grammar parsing by labelling the source-side alone. This is achieved through a novel training scheme where the labelling models are trained over the word-aligned bitext and gold-standard pruning labels are obtained by projecting target-side constituents to the source words. To our knowledge, this is the first work to apply this technique to MT decoding.

The proposed method is easy to implement and effective in practice. Results on a full-scale English-to-German experiment show that it gives more than 60% speed-up over a strong cube pruning baseline, with no loss in BLEU. While we use a string-to-tree model in this paper, the approach can be adapted to other syntax-based models.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 352–357,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics



r_1	KON	→	⟨ but, denn ⟩
r_2	NP-SB	→	⟨ we, wir ⟩
r_3	NP-OA	→	⟨ that reform process, diesen Reformprozeß ⟩
r_4	TOP	→	⟨ X$_1$. , S-TOP$_1$. ⟩
r_5	S-TOP	→	⟨ but X$_1$ need X$_2$, denn NP-OA$_2$ brauchen NP-SB$_1$ ⟩

Figure 1: A selection of grammar rules extractable from an example word-aligned sentence pair.

2 The Baseline String-to-Tree Model

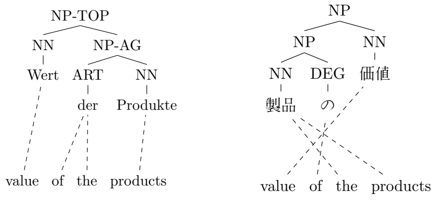
Our baseline translation model uses the rule extraction algorithm of Chiang (2007) adapted to a string-to-tree grammar. After extracting phrasal pairs using the standard approach of Koehn et al. (2003), all pairs whose target phrases are not exhaustively dominated by a constituent of the parse tree are removed and each remaining pair, $\langle \overline{f}, \overline{e} \rangle$, together with its constituent label, C, forms a lexical grammar rule: $C \rightarrow \langle \overline{f}, \overline{e} \rangle$. The rules r_1, r_2, and r_3 in Figure 1 are lexical rules. Non-lexical rules are generated by eliminating one or more pairs of terminal substrings from an existing rule and substituting non-terminals. This process produces the example rules r_4 and r_5.

Our decoding algorithm is a variant of CKY and is similar to other algorithms tailored for specific syntactic translation grammars (DeNero et al., 2009; Hopkins and Langmead, 2010). By taking the source-side of each rule, projecting onto it the non-terminal labels from the target-side, and weighting the grammar according to the model's local scoring features, decoding is a straightforward extension of monolingual weighted chart parsing. Non-local features, such as n-gram language model scores, are incorporated through cube pruning (Chiang, 2007).

3 Chart Pruning

3.1 Motivations

The abstract rules and large non-terminal sets of many syntactic MT grammars cause translation overgeneration at the span level and render decoding inefficient. Prior work on monolingual syntactic parsing has demonstrated that by excluding chart cells that are likely to violate constituent constraints, decoding efficiency can be improved with no loss in accuracy (Roark and Hollingshead, 2008). We consider a similar mechanism for syntactic MT decoding by prohibiting subtranslation generation for chart cells violating synchronous-grammar constraints.



(a) en-de (b) en-jp

Figure 2: Two example alignments. In (a) "the products" does not have a consistent alignment on the target side, while it does in (b).

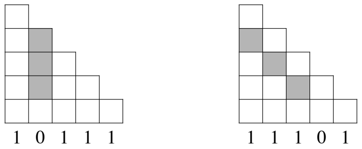
A motivating example is shown in Figure 2a, where a segment of an English-German sentence pair from the training data, along with its word alignment and target-side parse tree is depicted. The English phrases "value of" and "the products" do not have corresponding German translations in this example. Although the grammar may have rules to translate these two phrases, they can be safely pruned for this particular sentence pair.

In contrast to chart pruning for monolingual parsing, our pruning decisions are based on the source context, its target translation and the mapping between the two. This distinction is important since the syntactic correspondence between different language pairs is different. Suppose that we were to translate the same English sentence into Japanese (Figure 2a); unlike the English to German example, the English phrase "the products" will be a valid phrase that has a Japanese translation under a target constituent, since it is syntactically aligned to "製品" (Figure 2b).

The key question to consider is how to inject target syntax and word alignment information into our labelling models, so that pruning decisions can be based on the source alone, we address this in the following two sections.

3.2 Pruning by Labelling

We use binary tags to indicate whether a source word can start or end a multi-word phrase that has



(a) *b*-tags (b) *e*-tags

Figure 3: The pruning effects of two types of binary tags. The shaded cells are pruned and two types of tags are assigned independently.

a consistently aligned target constituent. We call these two types the *b*-tag and the *e*-tag, respectively, and use the set of values $\{0, 1\}$ for both.

Under this scheme, a *b*-tag value of 1 indicates that a source word can be the start of a source phrase that has a consistently aligned target phrase; similarly an *e*-tag of 0 indicates that a word cannot end a source phrase. If either the *b*-tag or the *e*-tag of an input phrase is 0, the corresponding chart cells will be pruned. The pruning effects of the two types of tags are illustrated in Figure 3. In general, 0-valued *b*-tags prune a whole column of chart cells and 0-valued *e*-tags prune a whole diagonal of cells; and the chart cells on the first row and the top-most cell are always kept so that complete translations can always be found.

We build a separate labeller for each tag type using gold-standard *b*- and *e*-tags, respectively. We train the labellers with maximum-entropy models (Curran and Clark, 2003; Ratnaparkhi, 1996), using features similar to those used for suppertagging for CCG parsing (Clark and Curran, 2004). In each case, features for a pruning tag consist of word and POS uni-grams extracted from the 5-word window with the current word in the middle, POS trigrams ending with the current word, as well as two previous tags as a bigram and two separate uni-grams. Our pruning labellers are highly efficient, run in linear time and add little overhead to decoding. During testing, in order to prevent over-pruning, a probability cutoff value θ is used. A tag value of 0 is assigned to a word only if its marginal probability is greater than θ.

3.3 Gold-standard Pruning Tags

Gold-standard tags are extracted from the word-aligned bitext used for translation model training, respecting rule extraction constraints, which is crucial for the success of our method.

For each training sentence pair, gold-standard *b*-tags and *e*-tags are assigned separately to the source words. First, we initialize both tags of each source word to 0s. Then, we iterate through all target constituent spans, and for each span, we find its corresponding source phrase, as determined by the word alignment. If a constituent exists for the phrase pair, the *b*-tag of the *first* word and the *e*-tag of the *last* word in the source phrase are set to 1s, respectively. Pseudocode is shown in Algorithm 1.

Algorithm 1 Gold-standard Labelling Algorithm

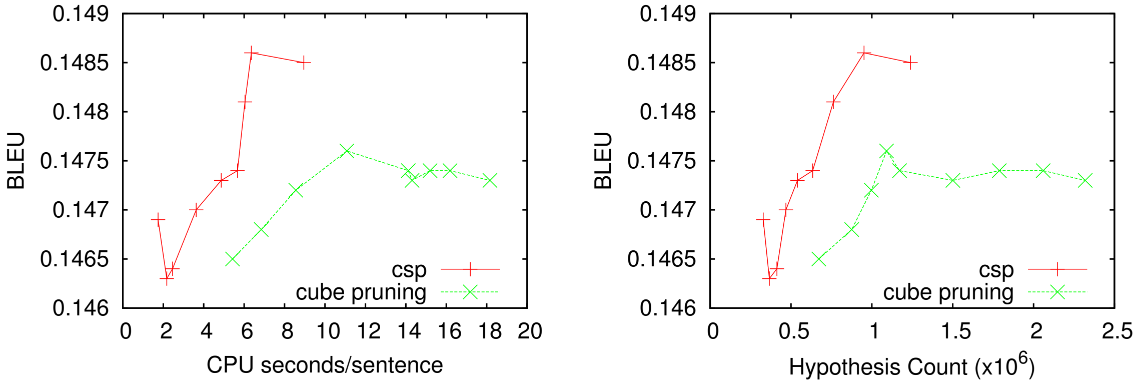
```
Input  forward alignment $A_{e\sim f}$, backward alignment $\hat{A}_{f\sim e}$ and 1-best parse tree $\tau$ for $f$
Output  Tag sequences $\boldsymbol{b}$ and $\boldsymbol{e}$ for e
1:  procedure TAG($e$, $f$, $\tau$, $A$, $\hat{A}$)
2:      $l \leftarrow |e|$
3:      for $i \leftarrow 0$ to $l - 1$ do
4:          $\boldsymbol{b}[i] \leftarrow 0$, $\boldsymbol{e}[i] \leftarrow 0$
5:      for $f[i', j']$ in $\tau$ do
6:          $\boldsymbol{s} \leftarrow \{\hat{A}[k] \mid k \in [i', j']\}$
7:          if $|\boldsymbol{s}| \leq 1$ then continue
8:          $i \leftarrow \min(\boldsymbol{s})$, $j \leftarrow \max(\boldsymbol{s})$
9:          if CONSISTENT($i, j, i', j'$) then
10:             $\boldsymbol{b}[i'] \leftarrow 1$, $\boldsymbol{e}[j'] \leftarrow 1$

11: procedure CONSISTENT($i, j, i', j'$)
12:     $\boldsymbol{t} \leftarrow \{A[k] \mid k \in [i, j]\}$
13:     return $\min(\boldsymbol{t}) \geq i'$ and $\max(\boldsymbol{t}) \leq j'$
```

Note that our definition of the gold-standard allows source-side labels to integrate bilingual information. On line 6, the target-side syntax is projected to the source; on line 9, consistency is checked against word alignment.

Consider again the alignment in Figure 2a. Taking the target constituent span covering “der Produkte” as an example, the source phrase under a consistent word alignment is “of the products”. Thus, the *b*-tag of “of” and the *e*-tag of “products” are set to 1s. After considering all target constituent spans, the complete *b*- and *e*-tag sequences for the source-side phrase in Figure 2a are $[1, 1, 0, 0]$ and $[0, 0, 1, 1]$, respectively. Note that, since we never prune single-word spans, we ignore source phrases under consistent one-to-one or one-to-many alignments.

From the gold standard data, we found 73.69% of the 54M words do not begin a multi-word aligned phrase and 77.71% do not end a multi-word aligned phrase; the 1-best accuracies of the two labellers tested on a held-out 20K sentences are 82.50% and 88.78% respectively.



(a) time vs. BLEU

(b) hypo count vs. BLEU

Figure 4: Translation quality comparison with the cube pruning baseline.

4 Experiments

4.1 Setup

A Moses (Koehn et al., 2007) string-to-tree system is used as our baseline. The training corpus consists of the English-German sections of the Europarl (Koehn, 2005) and the News Commentary corpus. Discarding pairs without target-side parses, the final training data has 2M sentence pairs, with 54M and 52M words on the English and German sides, respectively. Word-alignments are obtained by running GIZA++ (Och and Ney, 2000) in both directions and refined with “grow-diag-final-and” (Koehn et al., 2003). For all experiments, a 5-gram language model with Kneser-Ney smoothing (Chen and Goodman, 1996) built with the SRILM Toolkit (Stolcke and others, 2002) is used.

The development and test sets are the 2008 WMT newstest (2,051 sentences) and 2009 WMT newstest (2,525 sentences) respectively. Feature weights are tuned with MERT (Och, 2003) on the development set and output is evaluated using case-sensitive BLEU (Papineni et al., 2002). For both rule extraction and decoding, up to seven terminal/non-terminal symbols on the source-side are allowed. For decoding, the maximum span-length is restricted to 15, and the grammar is pre-filtered to match the entire test set for both the baseline system and the chart pruning decoder.

We use two labellers to perform b- and e-tag labelling independently prior to decoding. Training of the labelling models is able to complete in under 2.5 hours and the whole test set is labelled in under 2 seconds. A standard perceptron POS tagger (Collins, 2002) trained on Wall Street Journal sections 2-21 of the Penn Treebank is used to assign POS tags for both our training and test data.

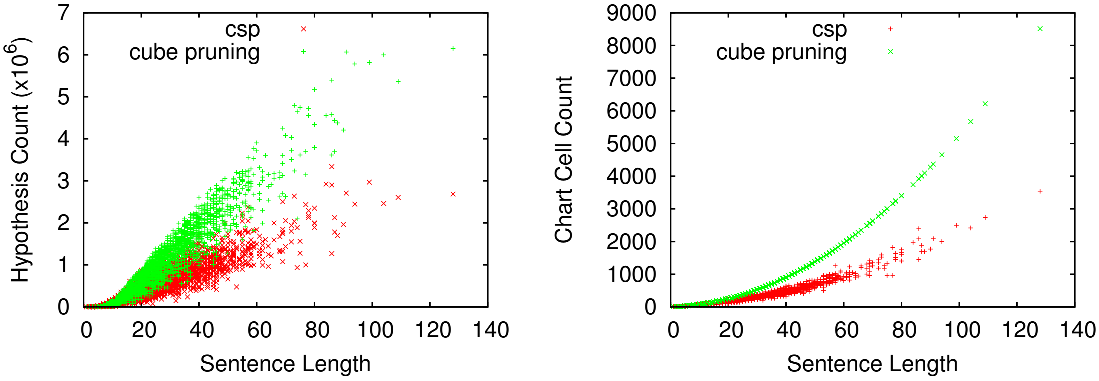
4.2 Results

Figures 4a and 4b compare CSP with the cube pruning baseline in terms of BLEU. Decoding speed is measured by the average decoding time and average number of hypotheses generated per sentence. We first run the baseline decoder under various beam settings ($b = 100$ - 2500) until no further increase in BLEU is observed. We then run the CSP decoder with a range of θ values ($\theta = 0.91 - 0.99$), at the default beam size of 1000 of the baseline decoder. The CSP decoder, which considers far fewer chart cells and generates significantly fewer subtranslations, consistently outperforms the slower baseline. It ultimately achieves a BLEU score of 14.86 at a probability cutoff value of 0.98, slightly higher than the highest score of the baseline.

At all levels of comparable translation quality, our decoder is faster than the baseline. On average, the speed-up gained is 63.58% as measured by average decoding time, and comparing on a point-by-point basis, our decoder always runs over 60% faster. At the θ value of 0.98, it yields a speed-up of 57.30%, compared with a beam size of 400 for the baseline, where both achieved the highest BLEU.

Figures 5a and 5b demonstrate the pruning power of CSP (θ = 0.95) in comparison with the baseline (beam size = 300); across all the cutoff values and beam sizes, the CSP decoder considers 54.92% fewer translation hypotheses on average and the minimal reduction achieved is 46.56%.

Figure 6 shows the percentage of spans of different lengths pruned by CSP (θ = 0.98). As ex-



(a) sentence length vs. hypo count

(b) sentence length vs. cell count

Figure 5: Search space comparison with the cube pruning baseline.

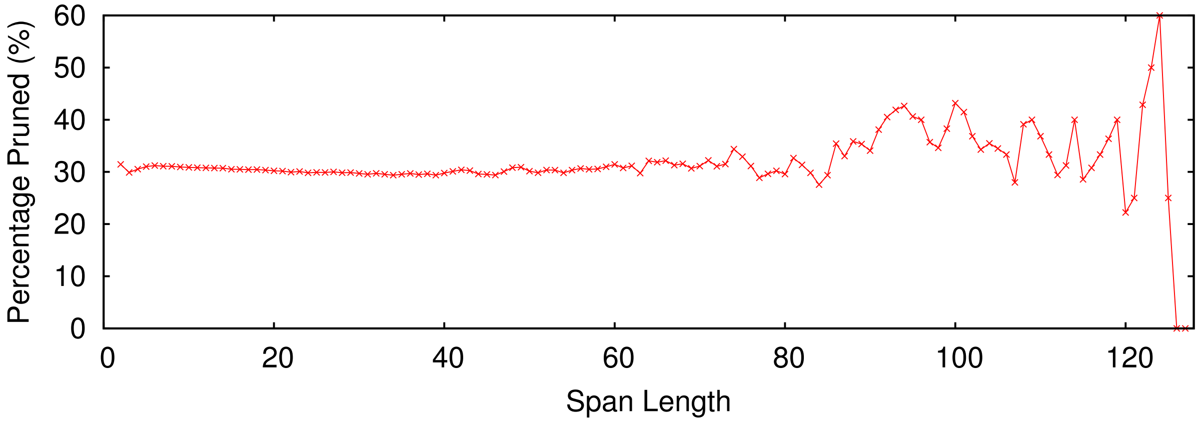


Figure 6: Percentage of spans of different lengths pruned at $\theta = 0.98$.

pected, longer spans are pruned more often, as they are more likely to be at the intersections of cells pruned by the two types of pruning labels, thus can be pruned by either type.

We also find CSP does not improve search quality and it leads to slightly lower model scores, which shows that some higher scored translation hypotheses are pruned. This, however, is perfectly desirable. Since our pruning decisions are based on independent labellers using contextual information, with the objective of eliminating unlikely subtranslations and rule applications. It may even offset defects of the translation model (i.e. high-scored bad translations). The fact that the output BLEU did not decrease supports this reasoning.

Finally, it is worth noting that our string-to-tree model does not force complete target parses to be built during decoding, which is not required in our pruning method either. We do not use any other heuristics (other than keeping singleton and the top-most cells) to make complete translation always possible. The hypothesis here is that good labelling models should not affect the derivation of complete target translations.

5 Conclusion

We presented a novel sequence labelling based, context-sensitive pruning method for a string-to-tree MT model. Our method achieves more than 60% speed-up over a state-of-the-art baseline on a full-scale translation task. In future work, we plan to adapt our method to models with different rule extraction algorithms, such as Hiero and forest-based translation (Mi and Huang, 2008).

Acknowledgements

We thank the anonymous reviewers for comments. The first author is fully supported by the Carnegie Trust and receives additional support from the Cambridge Trusts. Yue Zhang is supported by SUTD under the grant SRG ISTD 2012-038. Philip Williams and Philipp Koehn are supported under EU-FP7-287658 (EU BRIDGE).

References

S.F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. ACL*, pages 310–318.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

S. Clark and J.R. Curran. 2004. The importance of supertagging for wide-coverage ccg parsing. In *Proc. COLING*, page 282.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*, pages 1–8.

J.R. Curran and S. Clark. 2003. Investigating gis and smoothing for maximum entropy taggers. In *Proc. EACL*, pages 91–98.

John DeNero, Mohit Bansal, Adam Pauls, and Dan Klein. 2009. Efficient parsing for transducer grammars. In *Proc. NAACL-HLT*, pages 227–235.

M. Galley, M. Hopkins, K. Knight, and D. Marcu. 2004. What's in a translation rule. In *Proc. HLT-NAACL*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. COLING and ACL*, pages 961–968.

Kenneth Heafield, Philipp Koehn, and Alon Lavie. 2013. Grouping language model boundary words to speed k–best extraction from hypergraphs. In *Proc. NAACL.*

Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proc. EMNLP*, pages 646–655, October.

L. Huang and D. Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. ACL*, volume 45, page 144.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. NAACL-HLT*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL Demo Sessions*, pages 177–180.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, volume 5.

H. Mi and L. Huang. 2008. Forest-based translation rule extraction. In *Proc. EMNLP*, pages 206–214.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proc. ACL*, pages 440–447, Hongkong, China, October.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

S. Petrov, A. Haghighi, and D. Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proc. ACL*, pages 108–116.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. EMNLP*, volume 1, pages 133–142.

Brian Roark and Kristy Hollingshead. 2008. Classifying chart cells for quadratic complexity context-free inference. In *Proc. COLING*, pages 745–751.

Brian Roark and Kristy Hollingshead. 2009. Linear complexity context-free parsing pipelines via chart constraints. In *Proc. NAACL*, pages 647–655.

A. Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Proc. ICSLP*, volume 2, pages 901–904.

Hao Zhang and Daniel Gildea. 2008. Efficient multipass decoding for synchronous context free grammars. In *Proc. ACL.*

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proc. NAACL*, pages 256–263.

Y. Zhang, B.G. Ahn, S. Clark, C. Van Wyk, J.R. Curran, and L. Rimell. 2010. Chart pruning for fast lexicalised-grammar parsing. In *Proc. COLING*, pages 1471–1479.