PORTABILITY OF SYNTACTIC STRUCTURE FOR LANGUAGE MODELING

Ciprian Chelba

Microsoft Speech.Net / Microsoft Research
One Microsoft Way
Redmond, WA 98052
chelba@microsoft.com

ABSTRACT

The paper presents a study on the portability of statistical syntactic knowledge in the framework of the structured language model (SLM). We investigate the impact of porting SLM statistics from the Wall Street Journal (WSJ) to the Air Travel Information System (ATIS) domain. We compare this approach to applying the Microsoft rule-based parser (NLPwin) for the ATIS data and to using a small amount of data manually parsed at UPenn for gathering the intial SLM statistics. Surprisingly, despite the fact that it performs modestly in perplexity (PPL), the model initialized on WSJ parses outperforms the other initialization methods based on in-domain annotated data, achieving a significant 0.4% absolute and 7% relative reduction in word error rate (WER) over a baseline system whose word error rate is 5.8%; the improvement measured relative to the minimum WER achievable on the N-best lists we worked with is 12%.

1. INTRODUCTION

The structured language model uses hidden parse trees to assign conditional word-level language model probabilities. The model is trained in two stages: first the model parameters are intialized from a treebank and then an N-best EM variant is employed for reestimating the model parameters.

Assuming that we wish to port the SLM to a new domain we have four alternatives for initializing the SLM:

• manual annotation of sentences with parse structure. This is expensive, time consuming and requires linguistic expertise. Consequently, only a small amount of data could be annotated this way.

• parse the training sentences in the new domain using an automatic parser ([1], [2], [3]) trained on a domain where a treebank is available already

• use a rule-based domain-independent parser ([4])

• port the SLM statistics as intialized on the treebanked-domain. Due to the way the SLM parameter reestimation works, this is equivalent to using the SLM as an automatic parser trained on the treebanked-domain and then applied to the new-domain training data.

We investigate the impact of different intialization methods and whether one can port statistical syntactic knowledge from a domain to another. The second training stage of the SLM is invariant during the experiments presented here.

We show that one can successfuly port syntactic knowledge from the Wall Street Journal (WSJ) domain — for which a manual treebank [5] was developed (approximatively 1M words of text) — to the Air Travel Information System (ATIS) [6] domain. The choice for the ATIS domain was motivated by the fact that it is different enough in style and structure from the WSJ domain and there is a small amount of manually parsed ATIS data (approximatively 5k words) which allows us to train the SLM on in-domain hand-parsed data as well and thus make a more interesting comparison.

The remaining part of the paper is organized as follows: Section 2 briefly describes the SLM followed by Section 3 describing the experimental setup and results. Section 4 discusses the results and indicates future research directions.

2. STRUCTURED LANGUAGE MODEL OVERVIEW

An extensive presentation of the SLM can be found in [7]. The model assigns a probability $P(W, T)$ to every sentence W and its every possible binary parse T. The terminals of T are the words of W with POStags, and the nodes of T are annotated with phrase headwords and non-terminal labels. Let W be a sentence of length n words to which

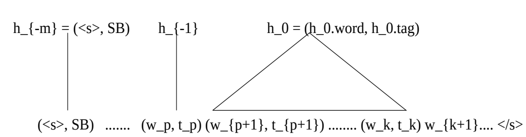

Fig. 1. A word-parse k-prefix

we have prepended the sentence begining marker `<s>` and appended the sentence end marker `</s>` so that $w_0 =$`<s>` and $w_{n+1} =$`</s>`. Let $W_k = w_0 \ldots w_k$ be the word k-prefix of the sentence — the words from the begining of

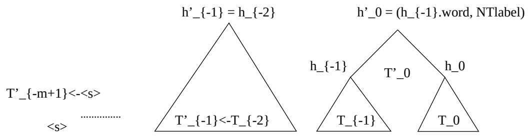

Fig. 2. Result of adjoin-left under NTlabel

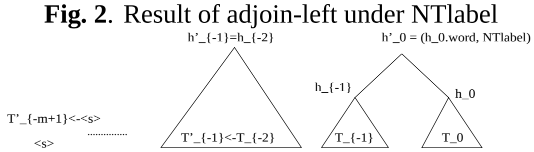

Fig. 3. Result of adjoin-right under NTlabel

the sentence up to the current position k — and W_kT_k the *word-parse k-prefix*. Figure 1 shows a word-parse k-prefix; `h_0 .. h_{-m}` are the *exposed heads*, each head being a pair (headword, non-terminal label), or (word, POStag) in the case of a root-only tree. The exposed heads at a given position k in the input sentence are a function of the word-parse k-prefix.

2.1. Probabilistic Model

The joint probability $P(W,T)$ of a word sequence W and a complete parse T can be broken into:

$$P(W,T) = \prod_{k=1}^{n+1}[\; P(w_k/W_{k-1}T_{k-1}) \cdot P(t_k/W_{k-1}T_{k-1}, w_k) \cdot \prod_{i=1}^{N_k} P(p_i^k/W_{k-1}T_{k-1}, w_k, t_k, p_1^k \ldots p_{i-1}^k)] \quad (1)$$

where:
- $W_{k-1}T_{k-1}$ is the word-parse $(k-1)$-prefix
- w_k is the word predicted by WORD-PREDICTOR
- t_k is the tag assigned to w_k by the TAGGER
- $N_k - 1$ is the number of operations the PARSER executes at sentence position k before passing control to the WORD-PREDICTOR (the N_k-th operation at position k is the `null` transition); N_k is a function of T
- p_i^k denotes the i-th PARSER operation carried out at position k in the word string; the operations performed by the PARSER are illustrated in Figures 2-3 and they ensure that all possible binary branching parses with all possible headword and non-terminal label assignments for the $w_1 \ldots w_k$ word sequence can be generated. The $p_1^k \ldots p_{N_k}^k$ sequence of PARSER operations at position k grows the word-parse $(k-1)$-prefix into a word-parse k-prefix.

Our model is based on three probabilities, each estimated using deleted interpolation and parameterized (approximated) as follows:

$$P(w_k/W_{k-1}T_{k-1}) = P(w_k/h_0, h_{-1}) \quad (2)$$
$$P(t_k/w_k, W_{k-1}T_{k-1}) = P(t_k/w_k, h_0, h_{-1}) \quad (3)$$
$$P(p_i^k/W_kT_k) = P(p_i^k/h_0, h_{-1}) \quad (4)$$

It is worth noting that if the binary branching structure developed by the parser were always right-branching and we mapped the POStag and non-terminal label vocabularies to a single type then our model would be equivalent to a trigram language model. Since the number of parses for a given word prefix W_k grows exponentially with k, $|\{T_k\}| \sim O(2^k)$, the state space of our model is huge even for relatively short sentences, so we had to use a search strategy that prunes it. Our choice was a synchronous multi-stack search algorithm which is very similar to a beam search.

The *language model* probability assignment for the word at position $k+1$ in the input sentence is made using:

$$P_{SLM}(w_{k+1}/W_k) = \sum_{T_k \in S_k} P(w_{k+1}/W_kT_k) \cdot \rho(W_k, T_k),$$
$$\rho(W_k, T_k) = P(W_kT_k) / \sum_{T_k \in S_k} P(W_kT_k) \quad (5)$$

which ensures a proper probability over strings W^*, where S_k is the set of all parses present in our stacks at the current stage k.

2.2. Model Parameter Estimation

Each model component — WORD-PREDICTOR, TAGGER, PARSER — is initialized from a set of parsed sentences after undergoing headword percolation and binarization. Separately for each model component we:
- gather counts from "main" data — about 90% of the training data
- estimate the interpolation coefficients on counts gathered from "check" data — the remaining 10% of the training data.

An N-best EM [8] variant is then employed to jointly reestimate the model parameters such that the PPL on training data is decreased — the likelihood of the training data under our model is increased. The reduction in PPL is shown experimentally to carry over to the test data.

3. EXPERIMENTS

We have experimented with three different ways of gathering the initial counts for the SLM — see Section 2.2:
- parse the training data (approximatively 76k words) using Microsoft's NLPwin and then intialize the SLM from these parse trees. NLPwin is a rule-based domain-independent parser developed by the natural language processing group at Microsoft [4].
- use the limited amount of manually parsed ATIS-3 data (approximatively 5k words)
- use the manually parsed data in the WSJ section of the Upenn Treebank. We have used the 00-22 sections (about 1M words) for initializing the WSJ SLM. The word vocabulary used for initializing the SLM on the WSJ data was

the ATIS open vocabulary — thus a lot of word types were mapped to the unknown word type.

After gathering the initial counts for all the SLM model components as described above, the SLM training proceeds in exactly the same way in all three scenarios. We reestimate the model parameters by training the SLM on the *same* training data (word level information only, all parse annotation information used for intialization is ignored during this stage), namely the ATIS-3 training data (approximatively 76k words), and using the *same* word vocabulary. Finally, we interpolate the SLM with a 3-gram model estimated using deleted interpolation:

$$P(\cdot) = \lambda \cdot P_{3gram}(\cdot) + (1 - \lambda) \cdot P_{SLM}(\cdot)$$

For the word error rate (WER) experiments we used the 3-gram scores assigned by the baseline back-off 3-gram model used in the decoder whereas for the perplexity experiments we have used a deleted interpolation 3-gram built on the ATIS-3 training data tokenized such that it matches the UPenn Treebank style.

3.1. Experimental Setup

The vocabulary used by the recognizer was re-tokenized such that it matches the Upenn vocabulary — e.g. *don't* is changed to *do n't*, see [7] for an accurate description. The re-tokenized vocabulary size was 1k. The size of the test set was 9.6k words. The OOV rate in the test set relative to the recognizer's vocabulary was 0.5%.

The settings for the SLM parameters were kept constant accross all experiments to typical values — see [7]. The interpolation weight between the SLM and the 3-gram model was determined on the check set such that it minimized the perplexity of the model initialized on ATIS manual parses and then fixed for the rest of the experiments.

For the speech recognition experiments we have used N-best hypotheses generated using the Microsoft Whisper speech recognizer [9] in a standard setup:

- feature extraction: MFCC with energy, one and two adjiacent frame differences respectively. The sampling frequency is 16kHz.
- acoustic model: standard senone-based, 2000 senones, 12 Gaussians per mixture, gender-independent models
- language model: Katz back-off 3-gram trained on the ATIS-3 training data (approximately 76k words)
- time-synchronous Viterbi beam search decoder

The N-best lists (N=30) are derived by performing an A^* search on the word hypotheses produced by the decoder during the search for the single best hypothesis. The 1-best WER —baseline — is 5.8% . The best achievable WER on the N-best lists generated this way is 2.1% — ORACLE WER — and is the lower bound on the SLM performance in our experimental setup.

3.2. Perplexity results

The perplexity results obtained in our experiments are summarized in Table 1. Judging on the initial perplexity of the stand-alone SLM ($\lambda = 0.0$), the best way to intialize the SLM seems to be by using the NLPwin parsed data; the meager 5k words of manually parsed data available for ATIS leads to sparse statistics in the SLM and the WSJ statistics are completely mismatched. However, the SLM iterative training procedure is able to overcome both these handicaps and after 13 iterations we end up with almost the same perplexity — within 5% relative of the NLPwin trained SLM but still above the 3-gram performance. Interpolation with the 3-gram model brings the perplexity of the trained models at roughly the same value, showing an overall modest 6% reduction in perplexity over the 3-gram model.

Initial Stats	Iter	λ = 0.0	λ = 0.6	λ = 1.0
NLPwin parses	0	21.3	16.7	16.9
NLPwin parses	13	17.2	15.9	16.9
SLM-atis parses	0	64.4	18.2	16.9
SLM-atis parses	13	17.8	15.9	16.9
SLM-wsj parses	0	8311	22.5	16.9
SLM-wsj parses	13	17.7	15.8	16.9

Table 1. Deleted Interpolation 3-gram + SLM; PPL Results

One important observation that needs to be made at this point is that although the initial SLM statistics come from different amounts of training data, all the models end up being trained on the same number of words — the ATIS-3 training data. Table 2 shows the number of distinct types (number of parameters) in the PREDICTOR and PARSER (see Eq. 2 and 4) components of the SLM in each training scenario. It can be noticed that the models end up having roughly the same number of parameters (iteration 13) despite the vast differences at initialization (iteration 0).

Initial Stats	Iter	PREDICTOR	PARSER
NLPwin parses	0	23,621	37,702
NLPwin parses	13	58,405	83,321
SLM-atis parses	0	2,048	2,990
SLM-atis parses	13	52,588	60,983
SLM-wsj parses	0	171,471	150,751
SLM-wsj parses	13	58,073	76,975

Table 2. Number of parameters for SLM components

3.3. N-best rescoring results

We have evaluated the models intialized in different conditions in a two pass — N-best rescoring — speech recognition setup. As can be seen from the results presented in Table 3 the SLM interpolated with the 3-gram performs best. The

SLM reestimation does not help except for the model initialized on the highly mismatched WSJ parses, in which case it proves extremely effective in smoothing out the SLM component statistics coming from out-of-domain. Not only is the improvement from the mismatched initial model large, but the trained SLM also outperforms the baseline and the SLM initialized on in-domain annotated data. We attribute this improvement to the fact that the initial model statistics on WSJ were estimated on a lot more data (more reliable) than the statistics coming from the little amount of ATIS data.

The SLM trained on WSJ parses achieved 0.4% absolute and 7% relative reduction in WER over the 3-gram baseline of 5.8%. The improvement relative to the minimum — ORACLE — WER achievable on the N-best list we worked with is in fact 12%. We have evaluated the statistical significance

Initial Stats	Iter	$\lambda = 0.0$	$\lambda = 0.6$	$\lambda = 1.0$
NLPwin parses	0	6.4	5.6	5.8
NLPwin parses	13	6.4	5.7	5.8
SLM-atis parses	0	6.5	5.6	5.8
SLM-atis parses	13	6.6	5.7	5.8
SLM-wsj parses	0	12.5	6.3	5.8
SLM-wsj parses	13	6.1	<u>5.4</u>	5.8

Table 3. Back-off 3-gram + SLM; WER Results

of the best result relative to the baseline using the standard test suite in the SCLITE package provided by NIST. The results are presented in Table 4. We believe that for WER statistics the most relevant significance test is the Matched Pair Sentence Segment one under which the SLM interpolated with the 3-gram is significant at the 0.003 level.

Test Name	p-value
Matched Pair Sentence Segment (Word Error)	0.003
Signed Paired Comparison (Speaker WER)	0.055
Wilcoxon Signed Rank (Speaker WER)	0.008
McNemar (Sentence Error)	0.041

Table 4. Significance Testing Results

4. CONCLUSIONS

The main conclusion that can be drawn is that the method for initializing the SLM is very important to the performance of the model. We consider this to be a promising venue for future research. The parameter reestimation technique proves extremely effective in smoothing the statistics coming from a different domain — mismatched initial statistics.

The syntactic knowledge embodied in the SLM statistics is portable but only in conjunction with the SLM parameter reestimation technique. The significance of this result lies in the fact that it is possible to use the SLM on a new domain where a treebank (be it generated manually or automatically) is not available.

5. ACKNOWLEDGEMENTS

Special thanks to Xuedong Huang and Milind Mahajan for useful discussions that contributed substantially to the work presented in this paper and for making available the ATIS N-best lists. Thanks to Eric Ringger for making available the NLPwin parser and making the necessary adjustments on its functionality.

6. REFERENCES

[1] Eugene Charniak, "A maximum-entropy-inspired parser," in *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139. Seattle, WA, 2000.

[2] Michael Collins, *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. thesis, University of Pennsylvania, 1999.

[3] Adwait Ratnaparkhi, "A linear observed time statistical parser based on maximum entropy models," in *Second Conference on Empirical Methods in Natural Language Processing*, Providence, R.I., 1997, pp. 1–10.

[4] George Heidorn, "Intelligent writing assistance," in *Handbook of Natural Language Processing*, R. Dale, H. Moisl, and H. Somers, Eds. Marcel Dekker, New York, 1999.

[5] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[6] P. Price, "Evaluation of spoken language systems: the ATIS domain," in *Proceedings of the Third DARPA SLS Workshop*, P. Price, Ed. Morgan Kaufmann, June 1990.

[7] Ciprian Chelba and Frederick Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, no. 4, pp. 283–332, 2000.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," in *Journal of the Royal Statistical Society*, vol. 39 of *B*, pp. 1–38. 1977.

[9] X. Huang et al., "From Sphinx-II to Whisper: Making speech recognition usable," in *Automated Speech and Speaker Recognition*, C. H. Lee, F. K. Soong, and K. K. Paliwal, Eds., pp. 481–588. Kluwer Academic, Norwell, MA, 1996.