Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 238-247. Association for Computational Linguistics.

The SuperARV Language Model: Investigating the Effectiveness of Tightly Integrating Multiple Knowledge Sources

Wen Wang and Mary P. Harper
School of Electrical and Computer Engineering
Purdue University
1285 The Electrical Engineering Building
West Lafayette, IN 47907-1285
{wang28,harper}@ecn.purdue.edu

Abstract

A new almost-parsing language model incorporating multiple knowledge sources that is based upon the concept of Constraint Dependency Grammars is presented in this paper. Lexical features and syntactic constraints are tightly integrated into a uniform linguistic structure called a *SuperARV* that is associated with a word in the lexicon. The SuperARV language model reduces perplexity and word error rate compared to trigram, part-of-speech-based, and parser-based language models. The relative contributions of the various knowledge sources to the strength of our model are also investigated by using constraint relaxation at the level of the knowledge sources. We have found that although each knowledge source contributes to language model quality, lexical features are an outstanding contributor when they are tightly integrated with word identity and syntactic constraints. Our investigation also suggests possible reasons for the reported poor performance of several probabilistic dependency grammar models in the literature.

1 Introduction

The purpose of a language model (LM) is to determine the *a priori* probability of a word sequence $w_1, \ldots, w_n$, $P(w_1, \ldots, w_n)$. Language modeling is essential in a wide variety of applications; we focus on speech recognition in our research. Although word-based LMs (with bigram and trigram being the most common) remain the mainstay in many continuous speech recognition systems, recent efforts have explored a variety of ways to improve LM performance (Niesler and Woodland, 1996; Chelba et al., 1997; Srinivas, 1997; Heeman, 1998; Chelba, 2000; Rosenfeld, 2000; Goodman, 2001; Roark, 2001; Charniak, 2001).

Class-based LMs attempt to deal with data sparseness and generalize better to unseen word sequences by first grouping words into classes and then using these classes to compute n-gram probabilities. Part-of-Speech (POS) tags were initially used as classes by Jelinek (1990) in a **conditional probabilistic model** (which predicts the tag sequence for a word sequence first and then uses it to predict the word sequence):

$$Pr(w_1^N) \approx \sum_{t_1,t_2,\ldots,t_N} \prod_{i=1}^{N} Pr(t_i|t_1^{i-1}) Pr(w_i|t_i) \quad (1)$$

However, Jelinek's POS LM is less effective at predicting word candidates than an n-gram word-based LM because it deletes important lexical information for predicting the next word. Heeman's (1998) POS LM achieves a perplexity reduction compared to a trigram LM by instead redefining the speech recognition problem as determining:

$$\begin{aligned} W^*, T^* &= arg \max_{W,T} P(W,T|A) \\ &= arg \max_{W,T} P(W,T)P(A|W,T) \\ &\approx arg \max_{W,T} P(W,T)P(A|W) \end{aligned}$$

where T is the POS sequence t_1^N associated with the word sequence $W = w_1^N$ given the speech utterance A. The LM $P(W,T)$ is a **joint probabilistic model** that accounts for both the sequence of words w_1^N and their tag assignments t_1^N by estimating the joint probabilities of words and tags:

$$P(w_1^N, t_1^N) = \prod_{i=1}^{N} P(w_i, t_i|w_1^{i-1}, t_1^{i-1}) \quad (2)$$

Johnson (2001) and Lafferty et al. (2001) provide insight into why a joint model is superior to a conditional model.

Recently, there has been good progress in developing structured models (Chelba, 2000; Charniak,

2001; Roark, 2001) that incorporate syntactic information. These LMs capture the hierarchical characteristics of a language rather than specific information about words and their lexical features (e.g., case, number). In an attempt to incorporate even more knowledge into a structured LM, Goodman (1997) has developed a probabilistic feature grammar (PFG) that conditions not only on structure but also on a small set of grammatical features (e.g., number) and has achieved parse accuracy improvement. Goodman's work suggests that integrating lexical features with word identity and syntax would benefit LM predictiveness. PFG uses only a small set of lexical features because it integrates those features at the level of the production rules, causing a significant increase in grammar size and a concomitant data sparsity problem that preclude the addition of richer features. This sparseness problem can be addressed by associating lexical features directly with words.

We hypothesize that high levels of word prediction capability can be achieved by tightly integrating structural constraints and lexical features at the word level. Hence, we develop a new dependency-grammar almost-parsing LM, *SuperARV LM*, which uses enriched tags called *SuperARVs*. In Section 2, we introduce our SuperARV LM. Section 3 compares the performance of the SuperARV LM to other LMs. Section 4 investigates the knowledge source contributions by constraint relaxation. Conclusions appear in Section 5.

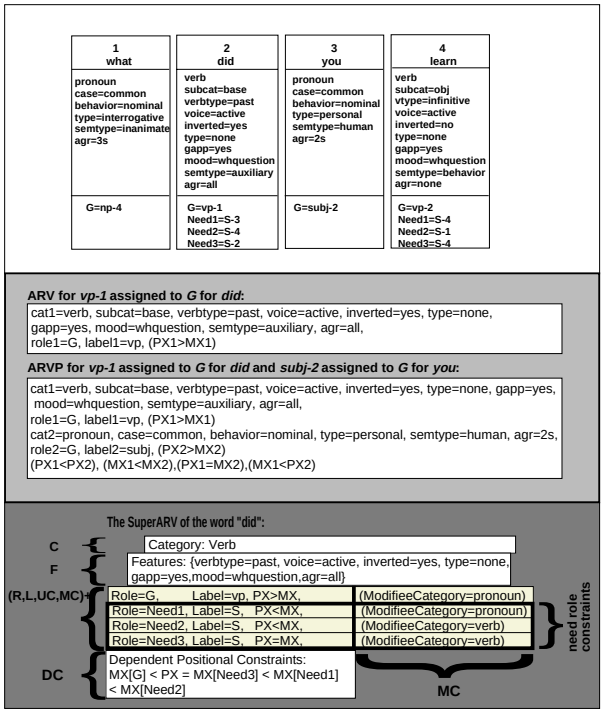

Figure 1: An example of a CDG parse, an ARV and ARVP, and the SuperARV of the word *did* in the sentence *what did you learn*. Note: *G* represents the governor role; the need roles, *Need1*, *Need2*, and *Need3*, are used to ensure that the grammatical requirements of the word are met. PX and MX([R]) represent the position of a word and its modifiee (for role R), respectively.

2 SuperARV Language Model

The SuperARV LM is a highly lexicalized probabilistic LM based on the Constraint Dependency Grammar (CDG) (Harper and Helzerman, 1995). CDG represents a parse as assignments of dependency relations to functional variables (denoted *roles*) associated with each word in a sentence. Consider the parse for *What did you learn* depicted in the white box of Figure 1. Each word in the parse has a lexical category and a set of feature values. Also, each word has a governor role (denoted G) which is assigned a role value, comprised of a label as well as a modifiee, which indicates the position of the word's governor or head. For example, the role value assigned to the governor role of *did* is `vp-1`, where its label `vp` indicates its grammatical function and its modifiee `1` is the position of its head *what*. The need roles (denoted $N1$, $N2$, and $N3$) are used to ensure the grammatical requirements (e.g., subcategorization) of a word are met, as in the case of the verb *did*, which needs a subject and a base form verb (but since the word takes no other complements, the modifiee of the role value assigned to N3 is set equal to its own position). Including need roles also provides a mechanism for using non-headword dependencies to constrain parse structures, which Bod (2001) has shown contributes to improved parsing accuracy.

During parsing, the grammaticality of a sentence in a language defined by a CDG is determined by applying a set of constraints to the possible role value assignments (Harper and Helzerman, 1995; Maruyama, 1990). Originally, the constraints were comprised of a set of hand-written rules specifying which role values (unary constraints) and pairs of role values (binary constraints) were grammatical (Maruyama, 1990). In order to derive the constraints directly from CDG annotated sentences, we have developed an algorithm to extract grammar relations using information derived directly from annotated sentences (Harper et al., 2000; Harper and Wang, 2001). Using the relationship between a role value's position and its modifiee's position, unary and binary constraints can be represented as a finite set of *abstract role values* (ARVs) and abstract role value pairs (ARVPs), respectively. The light gray box of Figure 1 shows an example of an ARV and an ARVP.

The ARV for the governor role value of *did* indicates its lexical category, lexical features, role, label, and positional relation information. ($PX1 > MX1$) indicates that *did* is governed by a word that precedes it. Note that the constraints of a CDG can be extracted from a corpus of parsed sentences.

A super abstract role value (*SuperARV*) is an abstraction of the joint assignment of dependencies for a word, which provides a mechanism for lexicalizing CDG parse rules. The dark gray box of Figure 1 presents an example of a SuperARV for the word *did*. The SuperARV structure provides an explicit way to organize information concerning one consistent set of dependency links for a word that can be directly derived from its parse assignments. SuperARVs encode lexical information as well as syntactic and semantic constraints in a uniform representation that is much more fine-grained than POS. A SuperARV can be thought of as providing admissibility constraints on syntactic and lexical environments in which a word may be used.

A SuperARV is formally defined as a four-tuple for a word, $\langle C, F, (R, L, UC, MC)+, DC\rangle$, where C is the lexical category of the word, $F = \{Fname_1 = Fvalue_1, \ldots, FName_f = FValue_f\}$ is a feature vector (where $Fname_i$ is the name of a feature and $Fvalue_i$ is its corresponding value), (R, L, UC, MC)+ is a list of one or more four-tuples, each representing an abstraction of a role value assignment, where R is a role variable, L is a functionality label, UC represents the relative position relation of a word and its dependent, MC is the lexical category of the modifiee for this dependency relation, and DC represents the relative ordering of the positions of a word and all of its modifiees. The following features are used in our SuperARV LM: `agr`, `case`, `vtype` (e.g., progressive), `mood`, `gapp` (e.g., gap or not), `inverted`, `voice`, `behavior` (e.g., mass, count), `type` (e.g., interrogative, relative). These lexical features constitute a much richer set than the features used by the parser-based LMs in Section 1. Since Harper et al. (1999) found that enforcing *modifiee constraints* (e.g., the lexical categories of modifiees) in parsing results in efficient pruning, we also include the modifiee lexical category (MC) in our SuperARV structure to impose modifiee constraints.

Words typically have more than one SuperARV to indicate different types of word usage. The average number of SuperARVs for words of different lexical categories vary, with verbs having the greatest SuperARV ambiguity. This is mostly due to the variety of feature combinations and variations on complement types and positions. We have observed in several experiments that the number of SuperARVs does not grow significantly as training set size increases; the moderate-sized Resource Management corpus (Price et al., 1988) with 25,168 words produces 328 SuperARVs, compared to 538 SuperARVs for the 1 million word Wall Street Journal (WSJ) Penn Treebank set (Marcus et al., 1993), and 791 for the 37 million word training set of the WSJ continuous speech recognition task.

SuperARVs can be accumulated from a corpus annotated with CDG relations and stored directly with words in a lexicon, so we can learn their frequency of occurrence for the corresponding word. A SuperARV can then be selected from the lexicon and used to generate role values that meet their constraints. Since there are no large benchmark corpora annotated with CDG information[1], we have developed a methodology to automatically transform constituent bracketing found in available treebanks into CDG annotations. In addition to generating dependency structures by headword percolation (Chelba, 2000), our transformer also utilizes a rule-based method to determine lexical features and need role values for words, as described by Wang et al. (2001).

Our SuperARV LM estimates the joint probability of words w_1^N and their SuperARV tags t_1^N:

$$\begin{aligned} Pr(w_1^N t_1^N) &= \prod_{i=1}^{N} Pr(w_i t_i | w_1^{i-1} t_1^{i-1}) \\ &= \prod_{i=1}^{N} Pr(t_i | w_1^{i-1} t_1^{i-1}) \cdot Pr(w_i | w_1^{i-1} t_1^{i}) \\ &\approx \prod_{i=1}^{N} Pr(t_i | w_{i-2}^{i-1} t_{i-2}^{i-1}) \cdot Pr(w_i | w_{i-2}^{i-1} t_{i-2}^{i}) \end{aligned} \quad (3)$$

Notice we use a joint probabilistic model to enable the joint prediction of words and their SuperARVs so that word form information is tightly integrated at the model level. Our SuperARV LM does not encode the word identity directly at the data structure level as was done in (Galescu and Ringger, 1999) since this could cause serious data sparsity problems.

To estimate the probability distributions in Equation (3) from training data, we use recursive linear interpolation among probability estimations of different orders. Representing each multiplicand in Equation (3) as the conditional probability $\hat{P}(x|y_1, y_2, \ldots, y_n)$ where $y_1, y_2, \ldots, y_n$ belong to a mixed set of words and SuperARVs, the recursive linear interpolation is calculated as follows:

[1]We have annotated a moderate-sized corpus, DARPA Naval Resource Management (Price et al., 1988), with CDG parse relations as reported in (Harper et al., 2000; Harper and Wang, 2001).

$$\hat{P}_n(x|y_1, y_2, \ldots, y_n)$$
$$= \lambda(x, y_1, y_2, \ldots, y_n) \cdot P_n(x|y_1, y_2, \ldots, y_n)$$
$$+(1 - \lambda(x, y_1, y_2, \ldots, y_n)) \cdot \hat{P}_{n-1}(x|y_1, y_2, \ldots, y_{n-1})$$

where:

- $y_1, y_2, \ldots, y_n$ is the context of order n-gram to predict x;
- $P_n(x|y_1, y_2, \ldots, y_n)$ is the order n-gram maximum likelihood estimation.

Table 1 enumerates the n-grams and their order for the interpolation smoothing of the two distributions in Equation (3). The ordering was based on our hypothesis that n-grams with more fine-grained history information should be ranked higher in the n-gram list since that information should be more helpful for discerning word and SuperARVs based on their history. The SuperARV LM hypothesizes categories for out-of-vocabulary words using the leave-one-out technique (Niesler and Woodland, 1996).

Table 1: The enumeration and order of n-grams for smoothing the distributions in Equation (3).

n-grams	$\hat{P}(t_i\|w_{i-2}^{i-1}t_{i-2}^{i-1})$	$\hat{P}(w_i\|w_{i-2}^{i-1}t_{i-2}^{i})$
highest	$\hat{P}(t_i\|w_{i-2}^{i-1}t_{i-2}^{i-1})$	$\hat{P}(w_i\|w_{i-2}^{i-1}t_{i-2}^{i})$
	$\hat{P}(t_i\|w_{i-1}t_{i-2}^{i-1})$	$\hat{P}(w_i\|w_{i-2}^{i-1}t_{i-1}^{i})$
	$\hat{P}(t_i\|w_{i-2}^{i-1}t_{i-1})$	$\hat{P}(w_i\|w_{i-1}t_{i-2}^{i})$
	$\hat{P}(t_i\|t_{i-2}^{i-1})$	$\hat{P}(w_i\|w_{i-1}t_{i-1}^{i})$
	$\hat{P}(t_i\|w_{i-1}t_{i-1})$	$\hat{P}(w_i\|w_{i-1}t_i)$
	$\hat{P}(t_i\|t_{i-1})$	$\hat{P}(w_i\|t_{i-1}^{i})$
lowest	$\hat{P}(t_i)$	$\hat{P}(w_i\|t_i)$

In preliminary experiments, we compared several algorithms for smoothing the probability estimations for our SuperARV LM. The best performance was achieved by using the modified Kneser-Ney smoothing algorithm initially introduced in (Chen and Goodman, 1998) and adapting it by employing a heldout data set to optimize parameters, including cutoffs for rare n-grams, by using Powell's search (Press et al., 1988). Parameters are chosen to optimize the perplexity on a heldout set.

In order to compare our SuperARV LM with a word-based LM, we must use the following equation to calculate the word perplexity (PPL):

$$\text{PPL} = 2^{\text{En}} \tag{4}$$
$$\text{En} \approx -\frac{1}{N}\sum_{i=1}^{N} \log_2 \hat{P}(w_i|w_{i-2}^{i-1})$$
$$\approx -\frac{1}{N}\sum_{i=1}^{N} \log_2 \frac{\sum_{t_{i-2,i}} \hat{P}(w_i t_i|w_{i-2}^{i-1}t_{i-2}^{i-1})\hat{P}(w_{i-2}^{i-1}t_{i-2}^{i-1})}{\sum_{t_{i-2,i-1}} \hat{P}(w_{i-2}^{i-1}t_{i-2}^{i-1})}$$

Equation (4) is used by class-based LMs to calculate word perplexity (Heeman, 1998). Parser-based LMs use a similar procedure that sums over parses.

The SuperARV LM is most closely related to the almost-parsing-based LM developed by Srinivas (1997). Srinivas' LM, based on the notion of a *supertag*, the elementary structure of Lexicalized Tree-Adjoining Grammar, achieved a perplexity reduction compared to a conditional POS n-gram LM (Niesler and Woodland, 1996). By comparison, our LM incorporates dependencies directly on words instead of through nonterminals, uses more lexical features than the supertag LM, uses joint instead of conditional probability estimations, and uses modified Kneser-Ney rather than Katz smoothing.

3 Evaluating the SuperARV Language Model

Traditionally, the LM quality in speech recognition is evaluated on two metrics: perplexity and WER, with the former commonly selected as a less computationally expensive alternative. We carried out two experiments, one using the Wall Street Journal Penn Treebank (WSJ PTB), a text corpus on which perplexity can be measured and compared to other LMs, and the Wall Street Journal Continuous Speech Recognition (WSJ CSR) task, a speech corpus on which both perplexity and WER can be evaluated after LM rescoring. These two experiments compare our SuperARV LM to a baseline trigram, a POS LM that was implemented using Equation (3) (where for this model t represents POS tags instead of SuperARV tags) and modified Kneser-Ney smoothing (as used in the SuperARV LM), and one or more parser-based LMs. Additionally, we evaluate the performance of *a conditional probability SuperARV LM* (denoted cSuperARV) implemented following Equation (1) rather than Equation (3) to evaluate the importance of using joint probability estimations.

For the WSJ PTB task, we compare the SuperARV LMs to the parser LMs developed by Chelba (2000), Roark (2001), and Charniak (2001). Although Srinivas (1997) developed an almost-parsing supertag-based LM, we cannot compare his LM with the other LMs because he used a small non-standard

subset of the WSJ PTB[2] and a trainable supertag LM is unavailable. Because none of the parser LMs has been fully trained for the WSJ CSR task, it is essential that we retrain them for comparison. The availability of a trainable version of Chelba's model enables us to train and test on the CSR task; however, because we do not have access to a trainable version of Charniak's or Roark's LMs, they are not considered in the CSR task. Note that for lattice rescoring, however, Roark found that Chelba's model achieves a greater reduction on WER than his LM (Roark, 2001).

3.1 Evaluating on the WSJ PTB

To evaluate the perplexity of the LMs on the WSJ PTB task, we adopted the conventions of Chelba (2000), Roark (2001), and Charniak (2001) for preprocessing the data. The vocabulary is limited to the most common 10K words, with all words outside this vocabulary mapped to $\langle UNK \rangle$. All punctuation is removed and no case information is retained. All symbols and digits are replaced by the symbol *N*. Sections 0-20 (929,564 words) are used as the training set for collecting counts, sections 21-22 (73,760 words) as the development set for tuning parameters, and sections 23-24 (82,430 words) for testing.

The baseline trigram uses Katz back-off model with Good-Turing discounting for smoothing. The POS, cSuperARV, and SuperARV LMs were implemented as described previously. The results for the parser-based LMs were initially taken from the literature. The perplexity on the test set using each LM and their interpolation with the corresponding trigram (and the interpolation weight) are shown in the top six rows of Table 2.

As can be seen in Table 2, the SuperARV LM obtains the lowest perplexity of all of the LMs (and so it is depicted in bold face). The SuperARV LM achieves the greatest perplexity reduction of 29.19% compared to the trigram, with Charniak's interpolated trihead LM a close second at 24.91%. The cSuperARV LM is clearly inferior to the SuperARV LM, even after interpolation. This result highlights the value of tight coupling of word, lexical feature, and syntactic knowledge both at the data structure level (which is the same for the SuperARV and cSuperARV LMs) and at the probability model level (which is different).

Notice that the cSuperARV, Chelba's, Roark's, and Charniak's LMs obtain an improvement in performance when interpolated with a trigram; whereas, the POS LM and the SuperARV LM do not benefit from trigram interpolation[3]. To gain more insight into why a trigram is effectively interpolated with some, but not all, of the LMs, we calculate the correlation of the trigram with each LM. A standard correlation is calculated between the probabilities assigned to each test set sentence by the trigram LM and the LM in question. This technique has been used in (Wang et al., 2002) to identify whether two LMs can be effectively interpolated.

Since we have access to an executable version of Charniak's LM trained on the WSJ PTB (ftp.cs.brown.edu/pub/nlparser) and a trainable version of Chelba's LM, we are able to calculate their correlations with our trigram LM. Chelba's LM was retrained using more parameter reestimation iterations than in (Chelba, 2000) to optimize the performance. Table 2 shows the correlation between each of the executable LMs and the trigram LM. The POS LM has the highest correlation with the trigram, closely followed by the SuperARV LM. Because these two LMs tightly integrate the word information jointly with the tag distribution, the trigram information is already represented. In contrast, the cSuperARV LM and Chelba's and Charniak's parser-based LMs have much lower correlations, indicating they have much lower overlap with the trigram. Because the cSuperARV LM only uses weak word distribution information in probability estimations, it leaves room for the trigram LM to compensate for the lack of word knowledge. The correlations for the parser-based LMs suggest that they capture different aspects of the words' distributions in the language than the words themselves.

3.2 Evaluating on the WSJ CSR Task

Next we compare the effectiveness of using the trigram word-based, POS, cSuperARV, SuperARV, and Chelba's LMs in rescoring hypotheses generated by a speech recognizer. The training set of the WSJ CSR task is composed of the 1987-1989 files containing 37,243,300 words. The speech data for the training set is used for building the acoustic model; whereas, the parse trees for the training set are generated following the policy that if the context-free grammar constituent bracketing can be found in the WSJ PTB, it becomes the parse tree for the training sentence; otherwise, we use the corresponding tree in the BLLIP treebank (Charniak et al., 2000). Since WSJ CSR is a speech corpus, there is no punctuation or case information. All words outside the provided vocabulary are mapped to $\langle UNK \rangle$. Note that

[2]Using the same 180,000 word training and 20,000 word test set as (Srinivas, 1997), our SuperARV LM obtains a perplexity of 92.76, compared to a perplexity of 101 obtained by the supertag LM.

[3]In the remaining experiments, the POS LM and the SuperARV LM are not interpolated with a trigram.

	Perplexity			
LM	**3gram**	**Model**	**Intp (Weight)**	**r**
POS	167.14	142.55	142.55 (1.0)	0.95
SuperARV	167.14	**118.35**	118.35 (1.0)	0.92
cSuperARV	167.14	150.01	143.83 (0.65)	0.68
Chelba (2000)	167.14	158.28	148.90 (0.64)	N/A
Roark (2001)	167.02	152.26	137.26 (0.64)	N/A
Charniak (2001)	167.89	130.20	126.07 (0.64)	N/A
Chelba	167.14	153.76	147.70 (0.64)	0.73
Charniak	167.14	130.20	126.03 (0.64)	0.69

Table 2: Comparing perplexity results for each LM on the WSJ PTB test set. 3gram represents the word-based trigram LM, *Intp (weight)* the LM interpolated with a trigram (and the interpolation weight), and r the correlation value. N/A means *not available*.

the word-level tokenization of treebank texts differs from that used in the speech recognition task with the major differences being: numbers (e.g., "1.2%" versus "one point two percent"), dates (e.g., "Dec. 20, 2001" versus "December twentieth, two thousand one") , currencies (e.g., "$10.25" versus "ten dollars and twenty five cents"), common abbreviations (e.g., "Inc." versus "Incorporated"), acronyms (e.g., "I.B.M." versus "I. B. M."), hyphenated and period-delimited phrases (e.g., "red-carpet" versus "red carpet"), and contractions and possessives (e.g., "do n't" versus "don't"). The POS, parser-based, and SuperARV LMs are all trained using the text-based tokenization from the treebank. Hence, during testing, a transformation converts the output of the recognizer to a form compatible with the text-based tokenization (Roark, 2001) for rescoring.

For testing the LMs, we use the four available WSJ CSR evaluation sets: 1992 5K closed vocabulary (denoted *92-5k*) with 330 utterances and 5,353 words, 1993 5K closed vocabulary (*93-5k*) with 215 utterances and 3,849 words, 1992 20K open vocabulary (*92-20k*) with 333 utterances and 5,643 words, and 1993 20K (*93-20k*) with 213 utterances and 3,446 words. We also employ a development set for each vocabulary size: *93-5k-dt* (513 utterances and 8,635 words) and *93-20k-dt* (252 utterances and 4,062 words).

The trigram provided by LDC for the CSR task was used due to its high quality. Before evaluation, all the other LMs (i.e., the POS LM, the cSuperARV and SuperARV LMs, and Chelba's LM) are retrained on the training set trees for the CSR task. Parameter tuning for the LMs on each task uses the corresponding development set[4].

Perplexity Results Table 3 shows the perplexity results for each test set with the best result for each in bold face. The SuperARV LM yields the lowest perplexity, with Chelba's LM a close second. The perplexity reductions for the SuperARV LM over the trigram across the test sets are 53.19%, 53.63%, 34.33%, and 32.05%, which is even higher than on the WSJ PTB task. This is probably due to the fact that more training data was used for the CSR task (37 million words versus 1 million words).

LM	**92-5k**	**93-5k**	**92-20k**	**93-20k**
3gram	45.61	50.51	106.52	109.22
POS	44.21	30.26	98.79	96.64
cSuperARV	36.53	28.50	86.83	89.12
SuperARV	**21.35**	**23.42**	**69.95**	**74.22**
Chelba	23.92	25.07	77.16	79.37

Table 3: Comparing perplexity results for each LM on the WSJ CSR test sets.

Rescoring Lattices Next using the same LMs, we rescored the lattices generated by an acoustic recognizer built using HTK (Ent, 1997). For each test set sentence, we generated a word lattice. We tuned the parameters of the LMs using the lattices on the corresponding development sets to minimize WER. Lattices were rescored using a Viterbi search for each LM.

Table 4 shows the WER and sentence accuracy (SAC) after rescoring lattices using each LM, with the lowest WER and highest SAC for each test set presented in bold face. We also give the lattice WER/SAC which defines the best accuracy possible given perfect knowledge. As can be seen from Table 4, the SuperARV LM produces the best reduction in WER with Chelba's LM the second best. When rescoring lattices on the 92-5k, 93-5k, 92-20k, and 93-20k test sets, the SuperARV LM yields a relative WER reduction of 13.54%, 9.70%, 8.64%, and 3.12% compared to the trigram, respectively. SAC results are similar: the SuperARV LM achieves an **absolute** increase on SAC of 4.24%, 6.97%, 2.7%, and 3.75%, compared to the trigram. Note that Chelba's LM

[4]The interpolation weight for cSuperARV for lattice rescoring was 0.63 on the 5k tasks and 0.60 on the 20k tasks, and 0.68 and 0.65 for Chelba's LM, respectively.

tied once with the SuperARV LM on 93-20k SAC, but always obtained higher WER across the four test sets. Because Chelba's LM focuses on developing the complete parse structure for a word sequence, it enforces more strict pruning based on the entire sentence. As can be seen in Table 4, the cSuperARV LM, even when interpolated with a trigram LM, obtains a lower accuracy than our SuperARV LM. This result is consistent with the hypothesis that a conditional model suffers from *label bias* (Lafferty et al., 2001).

The WER reported by Chelba (2000) on the 93-20k test set was 13.0%. This WER is lower than what we obtained for Chelba's retrained LM on the same task. This disparity is due to the fact that a higher quality acoustic decoder was used in (Chelba, 2000), which is not available to us. We further compare the LMs on Dr. Chelba's 93-20K lattices kindly provided by him, with the rescoring results shown in the last column of Table 4. We observe that Chelba's retrained LM improves his original result, but the SuperARV LM still obtains a greater accuracy. Sign tests show that the differences between the accuracies achieved by the SuperARV LM and the trigram, POS, and cSuperARV LMs are statistically significant. Although there is no significant difference between the SuperARV LM and Chelba's LM, the SuperARV LM has a much lower complexity than Chelba's LM.

4 Investigating the Knowledge Source Contributions

Next, we attempt to explain the contrast between the encouraging results from our SuperARV LM and the reported poor performance of several probabilistic dependency grammar models, i.e., the traditional probabilistic dependency grammar (PDG) LM, the probabilistic link grammar (PLG) (Lafferty et al., 1992) LM, and Zeman's probabilistic dependency grammar model (ZPDG) (Hajic et al., 1998). ZPDG was evaluated on the Prague Dependency Treebank (Hajic, 1998) during the 1998 Johns Hopkins summer workshop (Hajic et al., 1998) and produced a much lower parsing accuracy (under 60%) than Collins' probabilistic context-free grammar parser (80%) (Collins, 1996). Fong et al. (1995) evaluated the probabilistic link grammar LM described in (Lafferty et al., 1992) on small artificial corpora and found that the LM has a greater perplexity than a standard bigram. Additionally, only a modest improvement on the bigram was achieved after Fong and Wu (1995) revised the model to make grammar rule learning feasible.

One possible reason for their poor performance, especially in the light of our SuperARV LM results, is that these probabilistic dependency grammar models do not utilize sufficient knowledge to achieve a high level of accuracy. The knowledge sources the SuperARV LM uses, represented as components of the structure shown in Figure 1, include: lexical category (denoted c), lexical features (denoted f), role label or link type information (denoted L), a governor role dependency relation constraint (R, L, UC) (denoted g), a set of need role dependency relation constraints $(R, L, UC)+$ (denoted n), and modifiee constraints represented as the lexical category of the modifiee for each role (denoted m). Table 5 summarizes the knowledge sources that each of the probabilistic dependency grammar models uses. To determine whether the poor performance of the three probabilistic dependency grammar models results from our hypothesis that they utilize insufficient knowledge, we will evaluate our SuperARV LM after eliminating those knowledge sources that are not used by each of these models. Additionally, we will evaluate the contribution of each of the knowledge sources to the predictiveness of our SuperARV LM.

We use the methodology of selectively ignoring different types of knowledge as constraints to evaluate the knowledge source contributions to our SuperARV LM, as well as to approximate the performance of the other probabilistic dependency grammar models. The framework of CDG, on which our SuperARV LM is built, allows constraints to be tightened by adding more knowledge sources or loosened by ignoring certain knowledge. The SuperARV structure inherits this capability from CDG; selective constraint relaxation is implemented by eliminating one or more knowledge source in $\mathcal{K} = \{c, f, L, g, n, m\}$ from the SuperARV structure. We have constructed nine different LMs based on reduced SuperARV structures denoted **SARV-k** (i.e., a SuperARV structure after removing k with $k \subseteq \mathcal{K}$), where $-k$ represents the deletion of a subset of knowledge types (e.g., f, mn, $cgmn$). Each model is described next.

Modifiee constraints potentially hamper grammar generality, and so we consider their impact by deleting them from the LM by using the **SARV-m** structure. Need roles are important for capturing the structural requirements of different types of words (e.g., subcategorization), and we investigate their effects by using the **SARV-n** structure. The model based on **SARV-L** is built to investigate the importance of link type information. We can investigate the contribution of the combination of m and n, fundamental to the enforcement of valency constraints, by using the **SARV-mn** structure. The model based on **SARV-f** is used to evaluate whether

LM	Our HTK Lattices				Chelba's 93-20k lattices
	92-5k	93-5k	92-20k	93-20k	
	WER(SAC)	WER(SAC)	WER(SAC)	WER(SAC)	WER(SAC)
3gram	4.43(61.52)	6.91(43.26)	11.11(36.94)	14.74(30.52)	13.72(36.18)
POS	3.92(64.85)	6.55(47.91)	10.58(38.14)	14.54(32.39)	13.51(37.96)
cSuperARV	3.89(65.15)	6.42(48.84)	10.51(38.44)	14.45(32.86)	13.32 (38.22)
SuperARV	**3.83(65.76)**	**6.24(50.23)**	**10.15(39.64)**	**14.28(34.27)**	**12.87(42.02)**
Chelba	3.85(65.45)	6.26(49.77)	10.19(39.34)	14.36(**34.27**)	12.93(40.48)
lattice accuracy	1.79(79.40)	2.16(73.95)	4.93(59.46)	6.65(52.11)	3.41 (68.86)

Table 4: Comparing WER and SAC after rescoring lattices using each LM on WSJ CSR 5k- and 20k- test sets.

knowledge source	PDG	PLG	ZPDG	SARV
word identity	✓	✓	lemma	✓
lexical category (c)	✓		✓	✓
lexical features (f)			morpho-logical features	✓
link type (L)	✓	✓		✓
link direction (UC)	✓	✓	✓	✓
valency (n)				✓
modifiee constraints (m)			✓	✓

Table 5: Knowledge sources used by the three probabilistic dependency grammar models compared to our SuperARV LM. Note link type is defined as L and link direction is defined as UC in the SuperARV structure.

LM	92-5k	93-5k	92-20k	93-20k
3gram	45.61	50.51	106.52	109.22
SARV-cgmn	45.58	48.37	102.00	104.59
ZPDG-SARV	45.50	47.98	101.89	104.21
POS	44.21	30.26	98.79	96.64
SARV-gmn	43.16	27.75	96.69	93.25
SARV-fmn	45.01	27.42	96.23	93.16
SARV-f	42.33	27.06	94.87	90.20
SARV-mn	40.38	26.96	90.23	89.54
SARV-n	35.02	26.08	87.32	88.04
SARV-L	28.76	25.71	82.45	84.82
SARV-m	26.86	25.58	80.24	83.12
SARV	**21.35**	**23.42**	**69.95**	**74.22**

Table 6: Comparing perplexity results for each LM on the WSJ CSR test sets. The LMs appear in decreasing order of perplexity.

lexical features improve or degrade LM quality. The model based on **SARV-fmn** is very similar to the standard probabilistic dependency grammar LM, in which only word, POS, link type, and link direction information is used for probability estimations. The model based on **SARV-gmn** uses a feature augmentation of POS, and the model based on **SARV-cgmn** uses lexical features only. Additionally, we built the model **ZPDG-SARV** to approximate ZPDG. Zeman's PDG (Hajic et al., 1998) differs significantly from our original SuperARV LM in that it ignores label information L and some lexical feature information (the morphological tags do not include some lexical features having influence on syntax, denoted *syntactic lexical features*, i.e., gapp, inverted, mood, type, case, voice), and does not enforce valency constraints (instead, the model only counts the number of links associated with a word without discriminating whether the links represent governing or linguistic structural requirements). Also, word identity information is not used, instead, the model uses a loose integration of a word's lemma and its morphological tag. Given this analysis, we built the model ZPDG-SARV based on a structure including lexical category, morphological features, and (G, UC, MC).

Table 6 shows the perplexity results on the WSJ CSR test sets ordered from highest to lowest for each test set, with the best result for each in bold face. The full SuperARV LM yields the lowest perplexity. We found that ignoring modifiee constraints (SARV-m) increases perplexity the least, and ignoring link type information (SARV-L) and need role constraints (SARV-n) are a little worse than that. Ignoring both knowledge sources (SARV-mn) should result in even greater degradation, which is verified by the results. However, ignoring lexical features (SARV-f) produces an even greater increase in perplexity than relaxing both m and n. The SARV-fmn, which is closest to the traditional probabilistic dependency grammar LM, shows fairly poor quality, not much better than the POS LM. One might hypothesize that lexical features individually contribute the most to the overall performance of the SuperARV LM. However, using this knowledge source by itself (SARV-cgmn) results in dramatic degradation on perplexity, in fact even worse than that of the POS LM, but still slightly better than the baseline trigram. However, as demonstrated by SARV-gmn, the constraints from lexical features are strengthened by combining them with POS. Given the de-

scriptions in Table 5, we can approximate PLG by a model based on a SuperARV structure eliminating f and m (which should have a quality between SARV-f and SARV-fmn). It is noticeable that without word identity information, syntactic lexical features, and valency constraints, the ZPDG-SARV LM performs worse than the POS-based LM and only slightly better than the LM based on SARV-cgmn. This suggests that ZPDG can be strengthened by incorporating more knowledge.

The same ranking of the performance of the LMs was obtained for WER/SAC after rescoring the lattices using each LM, as shown in Table 7. Our experiments with relaxed SuperARV LMs suggest likely methods for improving PDG, PLG, and ZPDG models. The tight integration of word identity, lexical category, lexical features, and structural dependency constraints is likely to improve their performance. Clearly the investigated knowledge sources are quite synergistic, and their tight integration achieves the greatest improvement on both perplexity and WER.

5 Conclusions

We have compared our SuperARV LM to a variety of LMs and found that it achieves both perplexity and WER reductions compared to a trigram, and despite the fact that it is an almost-parsing LM, it outperforms (or performs comparably to) the more complex parser-based LMs on both perplexity and rescoring accuracy. Additional experiments reveal that selecting a joint instead of a conditional probabilistic model is an important factor in the performance of our SuperARV LM. The SuperARV structure provides a flexible framework that tightly couples a variety of knowledge sources without combinatorial explosion. We found that although each knowledge source contributes to the performance of the LM, it is the tight integration of the word level knowledge sources (word identity, POS, and lexical features) together with the structural information of governor and subcategorization dependencies that produces the best level of LM performance. We are currently extending the almost-parsing SuperARV LM to a full parser-based LM.

6 Acknowledgments

This research was supported by Intel, Purdue Research Foundation, and National Science Foundation under Grant No. IRI 97-04358, CDA 96-17388, and BCS-9980054. We would like to thank the anonymous reviewers for their comments and suggestions. We would also like to thank Dr. Charniak, Dr. Chelba, and Dr. Srinivas for their help with this research effort. Finally, we would like to thank Yang Liu (Purdue University) for providing us with the WSJ CSR test set lattices.

References

R. Bod. 2001. What is the minimal set of fragments that achieves maximal parse accuracy? In *Proceedings of ACL'2001.*

E. Charniak, D. Blaheta, N. Ge, K. Hall, and M. Johnson. 2000. BLLIP WSJ Corpus. CD-ROM. Linguistics Data Consortium.

E. Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of ACL'2001.*

C. Chelba, F. Jelinek, and S. Khudanpur. 1997. Structure and performance of a dependency language model. In *Proceedings of Eurospeech*, volume 5, pages 2775–2778.

C. Chelba. 2000. *Exploiting Syntactic Structure for Natural Language Modeling.* Ph.D. thesis, Johns Hopkins University.

S. F. Chen and J. T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University, Computer Science Group.

M. J. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of ACL'1996*, pages 184–191.

Entropic Cambridge Research Laboratory, Ltd., 1997. *HTK: Hidden Markov Model Toolkit V2.1.*

E. W. Fong and D. Wu. 1995. Learning restricted probabilistic link grammars. Technical Report HKUST-CS95-27, University of Science and Technology, Clear Water Bay, Hong Kong.

L. Galescu and E. K. Ringger. 1999. Augmenting words with linguistic information for n-gram language models. In *Proceedings of Eurospeech.*

J. Goodman. 1997. Probabilistic feature grammars. In *Proceedings of the Fourth international workshop on parsing technologies.*

J. Goodman. 2001. A bit of progress in language modeling, extended version. Technical Report MSR-TR-2001-72, Microsoft Research, Redmond, WA.

J. Hajic, E. Brill, M. Collins, B. Hladka, D. Jones, C. Kuo, L. Ramshaw, O. Schwartz, C. Tillmann, and D. Zeman. 1998. Core natural language processing technology applicable to multiple languages – Workshop '98. Technical report, Johns Hopkins Univ.

J. Hajic. 1998. Building a syntactically annotated corpus: The Prague Dependency Treekbank. In *Issues of Valency and Meaning (Festschrift for Jarmila Panevova)*, pages 106–132. Carolina, Charles University, Prague.

LM	92-5k WER(SAC)	93-5k WER(SAC)	92-20k WER(SAC)	93-20k WER(SAC)
3gram	4.43(61.52)	6.91(43.26)	11.11(36.94)	14.74(30.52)
SARV-cgmn	4.11(62.12)	6.78(45.12)	10.92(37.24)	14.63(31.46)
ZPDG-SARV	4.11(62.44)	6.71(46.02)	10.92(37.24)	14.63(31.65)
POS	3.92(64.85)	6.55(47.91)	10.58(38.14)	14.54(32.39)
SARV-gmn	3.92(64.85)	6.52(47.91)	10.56(38.14)	14.51(32.39)
SARV-fmn	3.92(64.85)	6.50(48.37)	10.53(38.14)	14.51(32.39)
SARV-f	3.92(64.85)	6.47(48.37)	10.49(38.14)	14.45(32.86)
SARV-mn	3.92(64.85)	6.44(48.37)	10.47(38.44)	14.42(32.86)
SARV-n	3.89(65.15)	6.39(48.37)	10.40(38.74)	14.39(33.33)
SARV-L	3.85(65.15)	6.29(48.92)	10.32(39.04)	14.39(33.33)
SARV-m	3.85(65.15)	6.29(49.77)	10.24(39.34)	14.35(33.80)
SARV	**3.83(65.76)**	**6.24(50.23)**	**10.15(39.34)**	**14.28(34.27)**

Table 7: Comparing WER and SAC after rescoring lattices using each LM on WSJ CSR 5k- and 20k- test sets. The LMs appear in decreasing order of WER.

M. P. Harper and R. A. Helzerman. 1995. Extensions to constraint dependency parsing for spoken language processing. *Computer Speech and Language*, 9:187–234.

M. P. Harper and W. Wang. 2001. Approaches for learning constraint dependency grammar from corpora. In *Proceedings of the Grammar and Natural Language Processing Conference*, Montreal, Canada.

M. P. Harper, S. A. Hockema, and C. M. White. 1999. Enhanced constraint dependency grammar parsers. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing*.

M. P. Harper, C. M. White, W. Wang, M. T. Johnson, and R. A. Helzerman. 2000. Effectiveness of corpus-induced dependency grammars for post-processing speech. In *Proceedings of NAACL'2000*.

P. A. Heeman. 1998. POS tagging versus classes in language modeling. In *Proceedings of the 6th Workshop on Very Large Corpora, Montreal*.

F. Jelinek. 1990. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*. Morgan Kaufman Publishers, Inc., San Mateo, CA.

M. Johnson. 2001. Joint and conditional estimation of tagging and parsing models. In *Proceedings of ACL'2001*.

J. D. Lafferty, D. Sleator, and D. Temperley. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Proc. of AAAI Fall Symp. Probabilistic Approaches to Natural Language*, Cambridge, MA.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML'2001*.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

H. Maruyama. 1990. Structural disambiguation with constraint propagation. In *The Proceedings of ACL'1990*, pages 31–38.

T. R. Niesler and P. C. Woodland. 1996. A variable-length category-based N-gram language model. In *Proceedings of ICASSP*, volume 1, pages 164–167.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. *Numerical Recipes in C*. Cambridge University Press, Cambridge.

P. J. Price, W. Fischer, J. Bernstein, and D. Pallett. 1988. A database for continuous speech recognition in a 1000-word domain. In *Proceedings of ICASSP'1988*, pages 651–654.

B. Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88:1270–1278.

B. Srinivas. 1997. *Complexity of lexical descriptions and its relevance to partial parsing*. Ph.D. thesis, University of Pennsylvania.

W. Wang and M. P. Harper. 2001. Investigating probabilistic constraint dependency grammars in language modeling. Technical Report TR-ECE-01-4, Purdue University, School of Electrical Engineering.

W. Wang, Y. Liu, and M. P. Harper. 2002. Rescoring effectiveness of language models using different levels of knowledge and their integration. In *Proceedings of ICASSP'2002*.