RERANKING MACHINE TRANSLATION HYPOTHESES WITH STRUCTURED AND WEB-BASED LANGUAGE MODELS

Wen Wang, Andreas Stolcke, Jing Zheng

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA, USA
{wwang,stolcke,zj}@speech.sri.com

ABSTRACT

In this paper, we investigate the use of linguistically motivated and computationally efficient structured language models for reranking N-best hypotheses in a statistical machine translation system. These language models, developed from Constraint Dependency Grammar parses, tightly integrate knowledge of words, morphological and lexical features, and syntactic dependency constraints. Two structured language models are applied for N-best rescoring, one is an almost-parsing language model, and the other utilizes more syntactic features by explicitly modeling syntactic dependencies between words. We also investigate effective and efficient language modeling methods to use N-grams extracted from up to 1 teraword of web documents. We apply all these language models for N-best re-ranking on the NIST and DARPA GALE program[1] 2006 and 2007 machine translation evaluation tasks and find that the combination of these language models increases the BLEU score up to 1.6% absolutely on blind test sets.

***Index Terms*—** Statistical machine translation, N-best reranking, structured language model, web-based language modeling, smoothing

1. INTRODUCTION

The goal of statistical machine translation (SMT) is to find the best translation $\hat{e}_1^{\hat{I}} = \hat{e}_1 \ldots \hat{e}_i \ldots \hat{e}_{\hat{I}}$ of source language sentence $f_1^J = f_1 \ldots f_j \ldots f_J$ where

$$\begin{aligned} \hat{e}_1^{\hat{I}} &= \arg\max_{I, e_1^I} Pr(e_1^I | f_1^J) \\ &= \arg\max_{I, e_1^I} Pr(f_1^J | e_1^I) \cdot Pr(e_1^I) \end{aligned}$$

Instead of using this source-channel approach, the direct modeling of the posterior probability $Pr(e_1^I | f_1^J)$ can be computed as follows by using a log-linear model [1]:

$$Pr(e_1^I | f_1^J) = \frac{exp(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J))}{\sum_{e'^{I'}_1} exp(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, f_1^J))}$$

where λ_m's are the weights (or scaling factors) for the models denoted by feature functions $h_m(\cdot)$. Since we can ignore the denominator, which is a normalization factor and is constant for a source sentence f_1^J, the goal of translation is to find

$$\hat{e}_1^{\hat{I}} = \arg\max_{I, e_1^I} \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)$$

With the above approach, we can easily integrate additional models $h(\cdot)$ as new knowledge sources and train the weights either using the maximum entropy principle or optimizing them based on the final translation performance using a certain evaluation metric such as BLEU or word error rate (WER). In this paper, we will investigate the efficacy of adding various language model (LM) reranking scores as additional knowledge sources and optimize the weights using minimum error training.

There has been much effort recently in MT on adding syntactically motivated features. Och and others [2] investigated the efficacy of integrating syntactic structures into a state-of-the-art SMT system by introducing feature functions representing syntactic information and discriminatively training scaling factors on a set of development N-best lists. They obtained consistent and significant improvement from the implicit syntactic features produced by IBM model 1 scores, but rather small improvement from other syntactic features, ranging from shallow to deep parsing approaches. Recently, Hasan and others [3] observed promising improvement of MT performance in a reranking framework by using supertagging and lightweight dependency analysis, a link grammar parser, and a maximum entropy based chunk parser. They achieved up to 0.7% absolute increase on BLEU on C-Star'03 and IWSLT'04 tasks. In this paper, we investigate the efficacy of structured LMs, which integrate lexical features and syntactic constraints, in an N-best reranking framework for SMT.

We also explore the use of large LMs derived from world-wide-web data for SMT reranking. Prior work along these lines includes distributed language modeling for N-best reranking [4] as well as introducing a simple, inexpensive smoothing method that can work reasonably well on very large amounts of data [5]. In this paper we make use of efficient deleted interpolation smoothing to accommodate very large databases, and address the problem of modified Kneser-Ney (KN) smoothing [6] from incomplete N-gram distributions. The latter is required because most of our web data is provided in the form of N-gram corpora, without access to the raw data.

2. MT SYSTEM DESCRIPTION

SRI's 2007 GALE evaluation Arabic-to-English and Chinese-to-English translation systems consist of two passes of decoding. The first pass uses a hierarchical phrase decoder developed at SRI to perform integrated decoding with a standard 4-gram LM to generate N-best lists. The basic phrases and hierarchical rules were extracted from parallel corpora and word-alignments provided by RWTH Aachen University, similar to David Chiang's approach [7]. The second pass rescores the N-best lists using several LMs trained on different corpora and estimated in different ways. The scores are then combined in the log-linear modeling framework [1] along

[1]The goal of the GALE program is to develop computer software techniques to analyze, interpret, and distill information from speech and text in multiple languages. For processing languages other than English, machine translation is an important module in the pipeline.

with other features used in the SMT system, including rule probabilities $p(f|e)$, $p(e|f)$, lexical weights $pw(f|e)$, $pw(e|f)$[8], sentence length, and rule counts. We optimized the weights using the minimum error training method to maximize BLEU scores using Amoeba simplex search on N-best lists, which could easily be extended to other objective functions such as word error rate (WER) and translation error rate (TER). The weights were optimized on a development set (GALE dev07) and applied to the blind test set, NIST MT eval06 GALE portion (denoted eval06).

3. STRUCTURED LMS

Syntax-based translation models have been shown to capture better long-range word order difference between the source and target language, and to produce higher quality translations than the standard phrase-based models[7, 9]. However, in the language modeling aspect, even the state-of-the-art syntax-based systems are not yet able to utilize syntactic properties of languages; instead, they rely on the standard n-gram language models that capture only local features.

In this paper, we investigate the idea of using relatively loose coupling methods to access rich syntactic information for MT compared to syntax-based models, by using structured LMs for N-best reranking. In the area of automatic speech recognition (ASR), structured LMs have recently been shown to give significant improvement in recognition accuracy relative to traditional word n-gram models [10, 11]. In [12], we developed an almost-parsing, SuperARV-based language model within the Constraint Dependency Grammar (CDG) framework for speech recognition. In this work we extend the almost-parsing LM to MT N-best reranking applications and also investigate the use of a parser LM for reranking. Note that it is important for these structured LMs to be computationally efficient in order to be applied to N-best rescoring for MT tasks, which in general deals with large amounts of data.

3.1. Almost-parsing language model

3.1.1. *Summary of the model*

The SuperARV LM [12] is a highly lexicalized probabilistic LM based on Constraint Dependency Grammars (CDGs), with grammar rules factored at the word level. It tightly integrates multiple knowledge sources, for example, word identity, morphological features, lexical features that have synergy with syntactic analyses (e.g., gap propagation, mood), and syntactic and semantic constraints, at both the *knowledge representation level* and *model level*.

Knowledge representation level integration was achieved by introducing a linguistic structure, called a super abstract role value (*SuperARV*), to encode multiple knowledge sources in a uniform representation that is much more fine-grained than part-of-speech (POS). A SuperARV is an abstraction of the joint assignment of all dependencies for a word, formally defined as a four-tuple $\langle C, F, (R, L, UC, MC)+, DC\rangle$, where C is the lexical category of the word, $F = \{Fname_1 = Fvalue_1, \ldots, Fname_f = Fvalue_f\}$ is a feature vector ($Fname_i$ 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 (e.g., governor), L is a functionality label (e.g., np), UC represents the relative position relation of a word and its dependent (i.e., modifiee), 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. Hence, the SuperARV structure for a word provides admissibility constraints on syntactic and lexical environments in which it may be used.

The second type, model-level integration was accomplished by jointly estimating the probabilities of a sequence of words w_1^N and their SuperARV membership t_1^N:

$$\begin{aligned} P(w_1^N t_1^N) &= \prod_{i=1}^{N} P(w_i t_i | w_1^{i-1} t_1^{i-1}) \\ &= \prod_{i=1}^{N} P(t_i | w_1^{i-1} t_1^{i-1}) \cdot P(w_i | w_1^{i-1} t_1^{i}) \end{aligned}$$

We use this to enable the joint prediction of words and their SuperARVs so that word identity information is tightly integrated at the model level. The SuperARV LM is fundamentally a class-based LM using SuperARVs as classes. N-gram conditional probabilities are estimated as follows:

$$P(w_i | w_1^{i-1}) = \frac{\sum_{t_{1,i}} P(w_i t_i | w_1^{i-1} t_1^{i-1}) P(w_1^{i-1} t_1^{i-1})}{\sum_{t_{1,i-1}} P(w_1^{i-1} t_1^{i-1})}$$

Detail on resolving parameter estimation issues appear in [12]. The SuperARV language model generates almost-parses for input sentences, since it does not explicitly selects modifiees but simply determines directions and position relations of all dependency links for each word. The SuperARV LM is most closely related to the almost-parsing LM developed by Srinivas [13] based on the supertags that are the elementary structures of Lexicalized Tree-Adjoining Grammar. For training SuperARV LMs, we developed a methodology to automatically transform context-free grammar (CFG) constituent bracketing into CDG annotations [14]. Then SuperARVs can be extracted from CDG parses and word and SuperARV statistics can be estimated.

3.1.2. *Modeling numbers and punctuation*

When developing an almost-parsing language model for SMT N-best reranking, we modified the modeling of the SuperARV LM in two ways. First, to improve generalization and coverage, numbers are mapped to a macro word "$number" during preprocessing on the parallel data and other target language model training data. For training the structured language models in this work, we delayed this mapping procedure by first generating valid CFG parse trees on the original word formats of numbers, then mapping the numbers in the parse trees to "$number".

Second, unlike in ASR language modeling, punctuation is present in the MT N-best hypotheses. Punctuation provides important syntactic information for SMT and needs to be modeled. To this end, we categorize punctuation marks into sentence-final punctuation (i.e., period, question mark, exclamation) and intra-sentence punctuation, and discriminate between them in modeling dependencies between punctuation and word tokens. When generating the CDG annotations for punctuation from CFG parse trees, we use their surface format, such as "." and ",", as their lexical categories[2]. Punctuation tokens bear no lexical features, i.e., no F is defined for them in the SuperARV tuple. When defining dependency relations with other words, as is common in the parsing community [15], we attach punctuation marks as high as possible in the CFG parse trees. We select the root of a sentence to be the headword for sentence-final punctuation marks. For intra-sentence punctuation marks, we treat them similarly to coordination by defining the headword of the following phrase as the headword of the punctuation mark. When a

[2]We found that this fine-grained categorization produced better parsing performance compared to clustering all punctuation into a single class.

sentence is incomplete and an intra-sentence punctuation mark appears at the end of a phrase, instead of between phrases, we pick up the headword of the preceding phrase as the headword of this punctuation.

3.2. Parser LM

In [16], we developed a statistical full parser-based LM. Given a sentence $W = w_1 \dots w_i \dots w_n$, the LM combines SuperARV tagging and modifiee specification by predicting the SuperARV tag sequence S and the set of dependency relations D for S (either in a loose or a tight coupling scheme). Compared to the almost-parsing LM, this language model can utilize long-distance dependency constraints and subcategorization information for word predictions. We observed improvement on word prediction accuracy by strengthening syntactic constraints as reported on other structured language models [17]. However, this full parser based LM utilizes statistics of dependencies between all pairs of words, and hence the computational complexity is quite high. In this section, we present a new parser LM that is much more efficient.

3.2.1. Using the baseNP model

First, we explored the idea of using baseNPs in dependency descriptions for parses [15]. Given a sentence W, we generate the reduced sentence $\bar{W}$ for it by first marking all baseNPs and then reducing all baseNPs to their headwords. A baseNP (or minimal NP) is a non-recursive NP such that none of its child constituents are NPs [15]. For example, a sentence "Mr. Viken is chairman of the Elsevier N.V., the Dutch publishing group" will be reduced to "Viken is chairman of N.V., group". Note that words internal to baseNPs cannot modify words outside their baseNP so they do not contribute much to enforcing long-span dependency constraints for the parser LM. In this way, words internal to baseNPs are not used for training and computing dependency statistics (during rescoring), and hence the efficiency of the LM is significantly improved.

To identify baseNPs for an input sentence W, we apply the baseNP model which is essentially a tagger to tag the boundaries between words using tags from the set S, C, E, B, N (denoting whether the boundary is at the start of a baseNP, continues a baseNP, is at the end of a baseNP, is between two adjacent baseNPs, or is between two words neither of which belongs to any baseNPs, respectively). Given the gap between words w_{i-1} and w_i denoted G_i, similar to [15], we use the two words to the left and right of G_i and their POS tags for predicting the tag of G_i. Different from [15], instead of just using commas as baseNP delimiters, we consider whether there is an intra-sentence punctuation between the words by introducing the variable c_i ($c_i = 1$ when there is an intra-sentence punctuation mark between w_{i-1} and w_i, and $c_i = 0$ otherwise). The baseNP model estimates the probability of tagging baseNPs B for the sentence W as (define $P(G_1|w_1, t_1) = 1$):

$$P(B|W) = \prod_{i=1,n} P(G_i|w_{i-1}, t_{i-1}, w_i, t_i, c_i)$$

where t_i is the part-of-speech tag of word w_i. The probability estimation is smoothed similar to [15], since that method proved to be simple and effective.

3.2.2. Further simplification of the parser LM

Our full parser LM [16] is a probabilistic generative model. After introducing the baseNP modeling, we modified the full parser LM as follows. Given T as a dependency annotation for a sentence W, the probability of W based on this LM is computed as:

$$\begin{aligned} P(W) &= \sum_T P(W,T) = \sum_T P(W,S,B,D) \\ &= \sum_T P(D|B,S,W)P(B|S,W)P(S,W) \end{aligned}$$

With Viterbi approximations and making independence assumptions, we approximate $P(W)$ as:

$$P(W) \approx \hat{P}(D|\bar{S})\hat{P}(B|W)\hat{P}(S,W)$$

This means that for an input sentence W, we first search for the best SuperARV sequence and best baseNP sequence for it. Then based on the reduced sentence $\bar{W}$ and the corresponding SuperARV tags $\bar{S}$ for the reduced sentence, we search for the best dependency relations that maximizes $P(D|\bar{S})$ using stack decoding as described in [16]. Note that in practice this approximation should be reasonable since by examining a non-trivial numbers of examples, we found that most of the baseNPs are very well defined and that parses among the highest scoring parses for a sentence generally share identical or very similar baseNP sequences, which are either the best baseNP sequence or very close to the best baseNP sequence.

4. LMS FOR LARGE WEB DATA COLLECTIONS

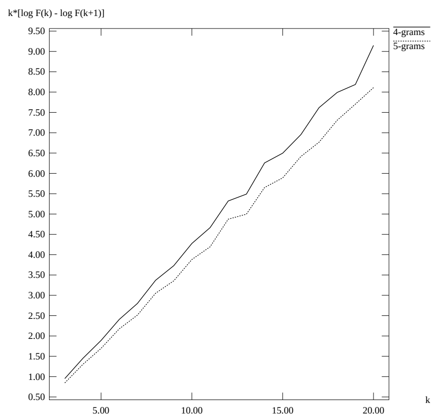
We trained two LMs that were based on N-gram corpora released by Google and Yahoo, respectively. The Google LM was based on 5-grams extracted from 1 terawords of web documents covering a wide range of sources. To limit the size of the distribution, only N-grams occurring 40 times or more were included; still it contained 0.31G bigrams, 0.98G trigrams, 1.3G 4-grams and 1.2G 5-grams, over a vocabulary of 11M words, and occupied about 25G on disk even after file compression. To build an LM that avoids memory issues we implemented a count-based LM representation in SRILM [18] using the Jelinek-Mercer deleted interpolation smoothing approach [6]. Interpolation weights were estimated on the held-out tuning set. During weight estimation and testing, the model needs access to only those N-grams occurring in the respective data sets, allowing for fast operation with limited memory. We refer to this type of LMs as a "count-LM".

The second web-based LM was trained on 5-grams provided by Yahoo, extracted from about 3.4G words of news sources during a 1-year period prior to the epoch of the GALE 2007 MT evaluation data. Although containing all N-grams occurring more than once, this collection was much smaller, but also more recent and directly focused on the target domain (broadcast news) than the Google corpus. It comprised about 54M bigrams, 187M trigrams, 288M 4-grams, and 296M 5-grams. The Yahoo LM we trained was a standard backoff LM using modified KN smoothing [6], for both 4-gram and 5-gram versions.

One interesting issue arising in building the Yahoo LM was that the number of singleton N-grams was not available in Yahoo's release (probably for the purpose of reducing the number of DVDs to store the data), and yet is required to compute the KN discounting values. By studying the distribution of N-grams in various corpora for different languages, we found an empirical law that seems to govern the progression of these counts of counts in large natural data sets:

$$\log F(k) - \log F(k+1) = \frac{\alpha}{k} \tag{1}$$

Fig. 1. Plot of count-of-count frequencies $F(k)$ according to the function $k[\log F(k) - \log F(k+1)]$. The reciprocal of the slope of the graph gives α from Equation 1 and allows extrapolation to unknown values of $F(k)$. In this case $\alpha = 2.42$ (4-grams) and 2.67 (5-grams) on Yahoo N-grams.



where $F(k)$ is the number of distinct N-grams of count k, and α is a constant dependent on the corpus and the N-gram order. Figure 1 demonstrates the regularity of the data by plotting the associated linear function. This law allows us to extrapolate from the available $F(k)$ to the missing ones, in this case, $F(1)$, by estimating the factor α based on the available counts of counts. After the extrapolation, we applied modified KN smoothing to train 4-gram and 5-gram LMs from the Yahoo data.

5. EXPERIMENTS AND RESULTS

The DARPA GALE program machine translation evaluation includes testing translation of text and machine transcription of recorded speech. The test includes language data from both Arabic and Chinese. The input to text MT is a variety of unstructured source language documents taken from newswire publications (denoted **NW**) and web-based newsgroups (denoted **WT**). The input to the recorded speech includes broadcast news (denoted **BN**) and broadcast conversations (talk and call-in shows, denoted **BC**). In the following experiments, we will investigate SMT performance on all of these four genres, NW, WT, BN, and BC, for both Arabic-English and Chinese-English.

5.1. Data

The word 4-gram LM used for search in the MT system was trained on various text sources with close to 5 billion tokens in total. We clustered them into the following 4 categories:

1: the English side of Arabic-English and Chinese-English parallel data provided by LDC (270 million tokens for CE and 280 million tokens for AE);

2: all of the English BN and BC transcriptions, web text, and translations for Mandarin and Arabic BN and BC, released under the DARPA EARS and GALE programs [3] (260 million tokens);

3: English Gigaword corpus (LDC2005T12, 2.5G tokens), North American News text corpora (LDC95T21 and LDC98T30, 620 million tokens);

4: Web data collected by SRI and BBN. SRI collected 30 million tokens of English news articles from June 2005 to January 2006. BBN's web data was collected by downloading articles from news web sites that offer free access to their archives, with total word counts as about 800 million tokens, and by crawling the "internet archive" and downloading past copies of pages from news websites, with total word counts as about 300 million tokens.

All training text was preprocessed following RWTH Aachen's treebank-style tokenization and text normalization toolkit, with some bug fixes and further cleanup especially for acoustic transcripts. The vocabulary of the word N-gram and structured language models is 2.3 million words, and we made sure all words appearing in the English side of the parallel data are covered. We trained separate word 4-gram LMs using modified KN smoothing on each of the four sources, and then created the final mixture LM by optimizing linear interpolation weights of these component LMs on a heldout tuning set. Note that we used a very high cutoff when training the component LM for the BBN web data, since they appear to be much more noisy compared to the LDC released news text. To compensate for this, we also trained a count-LM using all BBN web data without cutoff, denoting this LM as BBN-web-lm in the following experiments.

For the two structured language models, due to limited time for system development, we used only the first two sources for training the almost-parsing LM and the second source only for training the parser LM.

For N-best reranking, we used the word 4-gram LM (denoted **4-gram** or **4g**) (i.e., scores from search are used in combination), the almost-parsing LM (denoted **sarv**), the parser LM (denoted **plm**), the 5-gram count-LM trained on the Google N-grams (denoted **google**), the modified Kneser-Ney smoothed 4-gram LM trained on the Yahoo N-grams (denoted **yahoo**), and the 5-gram count-LM trained on all BBN web data (denoted **wlm**). Note that the perplexities from different language models using different vocabularies are not directly comparable, so here we only compared the perplexities from 4g, yahoo, sarv, and plm, on a 32K words LM tuning set. The perplexities for 4g, yahoo-4g, yahoo-5g, sarv, and plm are 126.99, 202.14, 196.76, 128.38, 280.64, respectively. Note that the two structured 5-gram LMs are trained on a small subset of the data used for training the word 4-gram, with the parser LM on an even smaller subset, but still the perplexity from the almost-parsing LM is almost the same as the word 4-gram LM.

5.2. Reranking experiments

For the AE and CE SMT tasks, we set the N-best list size to 3,000[4]. The word 4-gram is used in search and google, yahoo, sarv, plm, and wlm LMs are used for computing scores for each N-best hypothesis. The resulting LM scores are combined in a log-linear framework with weights optimized on the GALE dev07 test set and tested on the blind test set, NIST eval06 GALE subset in the 2006 NIST MT evaluation. Weights for knowledge sources are optimized on the combined set of NW and WT for the text MT, and BN and BC for the audio MT, respectively, to reduce the risk of over-fitting and dependence on accurate genre detections if doing optimization on the four genres, NW, WT, BN, and BC, separately. Note that for all of

[3] http://projects.ldc.upenn.edu/gale/data/catalog.html lists the LDC released corpora related to GALE.

[4] This nbest list size is selected for efficiency of MT decoding. We will experiment with decoding and rescoring with larger nbest lists in future work.

the experiments in this paper, we use BLEU scores based on one reference to measure MT performance.

Table 1. Effect of using Yahoo LMs with different training and rescoring schemes. BLEU sores [%] are measured for the GALE dev07 and eval06 Arabic-English BN and BC data. The baseline is to use the word 4-gram in search and the Google 5-gram count-LM for reranking.

LMs	**dev07 AE BLEU [%]**		**2006-nist/gale AE BLEU [%]**	
	BN	BC	BN	BC
Baseline(4g+google)	29.38	25.56	21.99	21.88
+yahoo-4g (KN)	29.87	25.86	22.34	22.35
+yahoo-5g (count-lm)	30.06	25.46	22.31	21.39
+yahoo-4g + Yahoo-5g (dynamic-interp, 0.6)	29.65	25.98	22.19	21.07
+yahoo-4g + yahoo-5g (static-interp)	29.57	26.07	22.59	21.72
+yahoo-4g + yahoo-5g (log-linear)	29.93	25.91	22.42	22.45

In the first experiment, we trained three variations of LMs using the Yahoo N-grams, i.e., a 4-gram LM using modified KN smoothing after extrapolation (yahoo-4g) and its 5-gram version (yahoo-5g), and a 5-gram count-LM using all of the Yahoo N-grams, denoted yahoo-5g (count-lm). The effect of using them for reranking dev07 and eval06 nbests is shown in Table 1. Note that on dev07 BN, the 5-gram count-LM using all of the Yahoo N-grams produced greater improvement than the modified KN smoothed Yahoo 4-gram, but the observation is reversed on BC. On the blind test set eval06, in fact yahoo-4g outperformed the 5-gram count-LM on both BN and BC. We compared the rescoring schemes for the modified KN smoothed Yahoo 4-gram and 5-gram, by using a dynamic (with weight as 0.6 for yahoo-4g) or static interpolation of them during rescoring, or using them in the log-linear framework. The picture is again different between BN and BC, on dev07 and eval06. On dev07 BN, the log-linear scheme yields the best improvement while on BC the winner is the static interpolation approach. On eval06, the static interpolation approach works best on BN but the log-linear scheme is best for BC. Since the gain in BLEU comes mostly from yahoo-4g (KN), for efficiency of reranking, we used yahoo-4g (KN) for reranking in the following experiments.

Table 2. Effect of using the almost-parsing LM (sarv) for N-best reranking, after adding google and yahoo LM scores. BLEU scores [%] are measured for the GALE dev07 and eval06 Arabic-English BN and BC data. Unsupervised adaptation is compared to static interpolation of component sarv LMs.

LMs	**dev07 AE BLEU [%]**		**2006-nist/gale AE BLEU [%]**	
	BN	BC	BN	BC
(1): 4g + google + yahoo	29.87	25.86	22.34	22.35
(2): (1) + dynamically interpolated sarv	30.59	25.45	22.27	22.52
(3): (1) + statically interpolated sarv	30.12	25.97	22.54	22.39

To further explore the advantages of the almost-parsing LM in capturing domain-specific grammatical features and word use, we investigated the effect of unsupervised adaptation. Note that we trained component SuperARV LMs after clustering sources in its training data and when using static interpolation, all of the component LMs are linearly interpolated with weights optimized on a LM tuning set to minimize its perplexity. We compared the reranking effect from statically or dynamically interpolating these component LMs. For dynamic interpolation, we computed the linear interpolation weights of component LMs by optimizing perplexities on the 1-best decoding hypothesis for each sentence. The results are shown in Table 2. Unsupervised adaptation produced improvement on dev07 BN but not on BC. On eval06, the observation is reversed. Still, one of the interpolation approaches yields improvement on BLEU over the baseline, 4g+google+yahoo. In future work, we will investigate possible factors contributing to this difference on performance of unsupervised adaptation, as well as effective adaptation approaches for the structured LMs.

Table 3. Effect of LM reranking on the dev07 test set for all the genres for AE.

LMs	**dev07 AE BLEU [%]**			
	NW	WT	BN	BC
(1) 4gram	31.20	22.99	28.72	25.22
(2) 4g+google	31.45	23.66	29.38	25.56
(3) 4g+yahoo	31.20	22.99	29.46	25.78
(4) 4g+google+yahoo	31.45	23.65	29.87	25.86
(5): (4)+sarv	31.71	23.59	30.59	25.45
(6): (4)+sarv+plm	32.15	23.99	**31.12**	26.30
(7): (4)+sarv+plm+wlm	**32.18**	**24.07**	31.04	**26.48**

Table 4. Effect of LM reranking on the blind test set, 2006 NIST MT evaluation Arabic-English GALE data.

LMs	**2006-nist/gale AE BLEU [%]**			
	NW	WT	BN	BC
(1) 4gram	27.36	15.59	21.67	21.58
(2) 4g+google	28.09	16.17	21.99	21.88
(3) 4g+yahoo	27.33	15.59	22.33	22.15
(4) 4g+google+yahoo	28.09	16.17	22.34	22.35
(5): (4)+sarv	28.01	16.26	22.27	22.52
(6): (4)+sarv+plm	**28.22**	16.16	23.01	**23.14**
(7): (4)+sarv+plm+wlm	28.11	**16.43**	**23.02**	23.05

Table 5. Effect of LM reranking on the dev07 test set for all the genres for CE.

LMs	**dev07 CE BLEU [%]**			
	NW	WT	BN	BC
(1) 4gram	16.72	14.63	19.62	16.34
(2) 4g+google	17.16	14.62	20.00	16.47
(3) 4g+yahoo	17.16	14.59	19.87	16.40
(4) 4g+google+yahoo	17.24	14.72	20.17	16.37
(5): (4)+sarv	17.40	14.89	20.44	16.27
(6): (4)+sarv+plm	17.51	15.11	20.56	16.70
(7): (4)+sarv+plm+wlm	**17.51**	**15.23**	**20.77**	**16.81**

Table 6. Effect of LM reranking on the blind test set, 2006 NIST MT evaluation Chinese-English GALE data.

LMs	2006-nist/gale CE BLEU [%]			
	NW	WT	BN	BC
(1): 4gram	17.71	14.16	17.17	15.51
(2): 4g+google	18.03	14.84	17.16	15.51
(3): 4g+yahoo	18.28	14.73	17.44	15.33
(4): 4g+google+yahoo	18.45	14.91	17.16	15.28
(5): (4)+sarv	**18.73**	**15.50**	17.51	15.55
(6): (4)+sarv+plm	18.61	15.09	17.56	15.95
(7): (4)+sarv+plm+wlm	18.72	15.04	**17.75**	**16.05**

Table 3, 4, 5, 6 present the BLEU scores from the baseline (i.e., no-reranking) and using various LMs for reranking on dev07 and eval06 for both Arabic-English and Chinese-English. On the Arabic-English dev07 test set, almost all of the LMs improve BLEU scores incrementally over the baseline 4-gram (or at least no degradation) and the combinations of all of the LMs for reranking yield the best improvement on all of the four genres with the exception of the BBN-web-lm on BN. The absolute improvement in BLEU scores ranges from 0.98% on NW, 1.08% on WT, 2.4% on BN, and 1.26% on BC. On the AE blind test set, 2006-nist/gale AE, the absolute improvement in BLEU scores from the combination of google and yahoo LMs ranges from 0.73% on NW, 0.58% on WT, 0.67% on BN, and 0.77% on BC. The combination of the two structured language models produced absolute improvement in BLEU scores of 0.67% on BN and 0.79% on BC. Adding the BBN-web-lm for reranking further helps improving the BLEU score on WT by 0.27% absolutely. The absolute gain from LM reranking on the AE blind test sets ranges from 0.86% on NW, 0.84% on WT, 1.35% on BN, and 1.56% on BC.

On the Chinese-English dev07 test set, the combination of all LMs yields the best improvement in BLEU scores on all of the four genres, ranging from 0.79% on NW, 0.6% on WT, 1.15% on BN, and 0.47% on BC. On the CE blind test set, the absolute gain from the combination of google and yahoo LMs ranges from 0.74% on NW, 0.75% on WT, no improvement on BN and BC (although the yahoo LM improves BN by 0.27%). Adding the almost-parsing LM improves the BLEU scores for each genre, up to 0.59% absolutely on WT. On the other hand, the parser LM helps especially for the BC genre with 0.4% absolutely. The BBN-web-lm also produced some small gains on all genres except WT. The combination of all of them tends to produce the best BLEU performance, except for the WT genre. The absolute gain from LM reranking on the CE blind test sets ranges from 1.02% on NW, 1.34% on WT, 0.58% on BN, and 0.54% on BC.

In all, for AE, these LMs for reranking help more on BN and BC; for CE, they help more on NW and WT. There are some fluctuations in BLEU scores when adding a new LM for reranking but in all, the combination of all these structured LMs and Yahoo and Google LMs tends to produce the best improvement, up to 1.6% absolutely on the blind test set. It is also noticeable that the patterns of gains from LMs are partially different or reversed between BN and BC, partly due to the significant difference between the two genres and the differences between the training data of these LMs and language features captured by them. Also, the patterns of gains are different between dev07 and eval06 in some cases, indicating there might be significant difference between dev07 and eval06 and the weight optimization approach could be improved to reduce the chance of getting trapped in a local optimum.

In future work, we will investigate genre-specific grammatical phenomena (e.g., speech disfluency for BC and the unstructured characteristics for WT) for structured LMs and effectiveness of adaptation approaches on them. We will also investigate more effective and robust approaches for combining multiple LMs for reranking as well as efficient approaches to employ more sophisticated LMs in search.

6. ACKNOWLEDGMENTS

We thank Richard Zens and Oliver Bender from RWTH Aachen University for making the MT training corpora and alignments available. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023 (approved for public release, distribution unlimited). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. REFERENCES

[1] F. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation", *in Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 295–302. Association for Computational Linguistics, 2002.

[2] F. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A smorgasbord of features for statistical machine translation", *in Proceedings of HLT-NAACL*, pp. 161–168, 2004.

[3] S. Hasan, O. Bender, and H. Ney, "Reranking translation hypotheses using structural properties", *in Proceedings of EACL*, 2006.

[4] Y. Zhang, A. S. Hildebrand, and S. Vogel, "Distributed language modeling for n-best list re-ranking", *in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 216–223, Sydney, Australia, July 2006. Association for Computational Linguistics.

[5] T. Brants, A. C. Popat, P. Xu, F. Och, and J. Dean, "Large language models in machine translation", *in Proceedings of Conference of Empirical Methods in Natural Language Processing*, pp. 858–867, 2007.

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

[7] D. Chiang, "A hierarchical phrase-based model for statistical machine translation", *in Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 263–270. Association for Computational Linguistics, 2005.

[8] P. Koehn, F. Och, and D. Marcu, "Statistical phrase-based translation", *in Proceedings of HLT-NAACL*, pp. 127–133, 2003.

[9] D. Marcu, W. Wang, A. Echihabi, and K. Knight, "SPMT: Statistical machine translation with syntactified target language phrases", *in Proceedings of Conference of Empirical Methods in Natural Language Processing*, pp. 44–52, 2006.

[10] C. Chelba and F. Jelinek, "Exploiting syntactic structure for language modeling", *in Proc. COLING-ACL*, vol. 1, pp. 225–231, Montreal, 1998.

[11] B. Roark, "Probabilistic top-down parsing and language modeling", *Computational Linguistics*, vol. 27, pp. 249–276, 2001.

[12] W. Wang and M. Harper, "The SuperARV language model:Investigating the effectiveness of tightly integrating multiple knowledge sources", *in Proceedings of Conference of Empirical Methods in Natural Language Processing*, 2002.

[13] B. Srinivas, *Complexity of lexical descriptions and its relevance to partial parsing*, PhD thesis, University of Pennsylvania, 1997.

[14] W. Wang, *Statistical Parsing and Language Modeling Based On Constraint Dependency Grammar*, PhD thesis, Purdue University, 2003.

[15] M. J. Collins, "A new statistical parser based on bigram lexical dependencies", *in Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 184–191, 1996.

[16] W. Wang and M. Harper, "Language modeling using a statistical dependency grammar parser", *in Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 519–524, 2003.

[17] P. Xu, C. Chelba, and F. Jelinek, "A study on richer syntactic dependencies for structured language modeling", *in In Proceedings of ACL 2002*, 2002.

[18] A. Stolcke, "SRILM—an extensible language modeling toolkit", in J. H. L. Hansen and B. Pellom, editors, *Proc. ICSLP*, vol. 2, pp. 901–904, Denver, Sep. 2002.