Statistical Machine Translation with Local Language Models

Christof Monz
Informatics Institute, University of Amsterdam
P.O. Box 94323, 1090 GH Amsterdam, The Netherlands
c.monz@uva.nl

Abstract

Part-of-speech language modeling is commonly used as a component in statistical machine translation systems, but there is mixed evidence that its usage leads to significant improvements. We argue that its limited effectiveness is due to the lack of lexicalization. We introduce a new approach that builds a separate local language model for each word and part-of-speech pair. The resulting models lead to more context-sensitive probability distributions and we also exploit the fact that different local models are used to estimate the language model probability of each word during decoding. Our approach is evaluated for Arabic- and Chinese-to-English translation. We show that it leads to statistically significant improvements for multiple test sets and also across different genres, when compared against a competitive baseline and a system using a part-of-speech model.

1 Introduction

Language models are an important component of current statistical machine translation systems. They affect the selection of phrase translation candidates and reordering choices by estimating the probability that an application of a phrase translation is a fluent continuation of the current translation hypothesis. The size and domain of the language model can have a significant impact on translation quality. Brants et al. (2007) have shown that each doubling of the training data from the news domain (used to build the language model), leads to improvements of approximately 0.5 BLEU points. On the other hand, each doubling using general web data leads to improvements of approximately 0.15 BLEU points.

While large n-gram language models do lead to improved translation quality, they still lack any generalization beyond the surface forms (Schwenk, 2007). Consider example (1), which is a short sentence fragment from the MT09 Arabic-English test set, with the corresponding machine translation output (1.b), from a phrase-based statistical machine translation system, and reference translation (1.c).

(1) a. ... خلفية تصريحات صحافية مثيرة للجدل واتهامهم له ...

b. ... the background of press statements of controversial and accused him ...

c. ... the background of controversial press statements and accused him ...

Clearly, the adjective "controversial" should precede the nouns "press statement", but since the AFP and Xinhua portions of the Gigaword corpus, used to build the language model for the translation system, do not contain this surface n-gram, translations with obviously ungrammatical constructions such as (1.b) can result. For unseen n-grams, one would like to model adjectives as being likely to precede nouns in English, for example.

A straightforward approach to address this is to exploit the part-of-speech (POS) tags of the target words during translation (Kirchhoff and Yang, 2005). Though models exploiting POS information are not expressive enough to model long-distance dependencies, they can account for locally ungrammatical constructions such as (1.b). Several attempts have been made to interpolate POS language models

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 869–879,
Edinburgh, Scotland, UK, July 27–31, 2011.

with surface models. Under constrained data conditions, this can lead to improvements. But once larger amounts of training data are used, the gains obtained from adding POS language models decline substantially. This raises the question of why POS language models are not more effective. We argue that one of the short-comings of previous approaches to using POS language models is that these models are estimated globally, not lexically anchored, and hence rather context insensitive.

In this paper, we introduce a novel approach that builds and uses individual, local POS language models for each word in the vocabulary. Our experiments show that it leads to statistically significant improvements over a competitive baseline, using lexicalized reordering and a sizable 5-gram word language model, as well as a standard 7-gram POS language model approach.

2 Part-of-Speech Language Models

2.1 Background

Typically, POS language models are used like word-based language models. N-grams are extracted from a POS-tagged corpus and an n-gram language model is built from that. While word-based models estimate the probability of a string of m words by Equation 2, POS-based models estimate the probability of string of m POS tags by Equation 3.

$$p(w_1^m) \propto \prod_{i=1}^{m} p(w_i|w_{i-n+1}^{i-1}) \quad (2)$$

$$p(t_1^m) \propto \prod_{i=1}^{m} p(t_i|t_{i-n+1}^{i-1}) \quad (3)$$

where, n is the order of the language model, and w_i^j refers to the sub-sequence of words (or tags) from positions i to j.

Word language models can be built directly from large text corpora, such as LDC's Gigaword corpus, but POS models require texts that are annotated with POS tags. Ideally, one would use manually annotated corpora such as the Penn Treebank (Marcus et al., 1993), but since those tend to be small, most approaches rely on larger corpora which have been automatically annotated by a POS tagger or a parser (Koehn et al., 2008). Though automated annotation inevitably contains errors, it is assumed that this is ameliorated by the increased size of annotated data.

The event space of a language models is of size $|V|^n$, where V is the vocabulary, and n is the order of the language model. The vocabulary of POS models, (typically ranging between 40 and 100 tags), is much smaller than the vocabulary of a word model, which can easily approach a million words. Nevertheless, most POS language modeling approaches apply some form of smoothing to account for unseen events (Bonneau-Maynard et al., 2007).

To deploy POS language models in machine translation, translation candidates need to be annotated with POS tags. Each target phrase $\bar{e}$ in a phrase pair $(\bar{f}, \bar{e})$ can be associated with a number of POS tag sequences $\bar{t}_{\bar{e}}$. Heeman (1998) shows that using the joint probability leads to improved perplexity for POS models. For machine translation one can sum over all possible tag sequences, as in Equation 4.

$$p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} \sum_{\mathbf{t}} p(\mathbf{e}, \mathbf{t}|\mathbf{f}) \quad (4)$$

Summing over all possible tag sequences has the disadvantage that it requires one to keep this information during decoding. Below, we opt for an approximate solution, where each target phrase is annotated with the most likely POS tag sequence given the source and target phrase: $\bar{t}_{\bar{e}} = \arg\max_{\bar{t}} p(\bar{t}|\bar{e}, \bar{f})$.

2.2 Effectiveness of POS Language Models

Reported results on the effectiveness of POS language models for machine translation are mixed, in particular when translating into languages that are not morphologically rich, such as English. While they rarely seem to hurt translation quality, there does not seem to be a clear consensus that they significantly improve quality either.

Koehn and Hoang (2007) have reported an increase of 0.86 BLEU points for German-to-English translation for small training data. After relaxing phrase-matching to include lemma and morphological information on the source side, POS language models lead to a decrease of -0.42 BLEU points. Supertagging encapsulates more contextual information than POS tags and Birch et al. (2007) report improvements when comparing a supertag language model to a baseline using a word language model

only. Once the baseline incorporates lexicalized distortion (Tillmann, 2004; Koehn et al., 2005), these improvements disappear. Factored language models have not resulted in significant improvements either. Kirchhoff and Yang (2005) report slight improvements when re-ranking the n-best lists of their decoder, which word tri-grams. But these improvements are less than those gained by re-ranking the n-best lists with a 4-gram word language model.

The impact of POS language models depends among other things on the size of the parallel corpus, the size and order of the word language model, and whether lexicalized distortion models are used. To gauge the potential effectiveness of POS language models without taking into consideration all these factors, we isolate the contribution of the language model by simulating machine translation output using English data only (Al-Onaizan and Papineni, 2006; Post and Gildea, 2008). Taking a set of POS-tagged reference translations of the MT04 Arabic-to-English test set, each English sentence is randomly chunked into n-grams of average length three. The chunks of each sentence, with their corresponding POS tags, are randomly reordered. This is repeated 500 times for each sentence in the test set. The smoothed sentence BLEU score (ignoring brevity penalty) is computed for each reordered sentence with respect to all reference translations. The higher the BLEU score, the more well-formed the reordering is. As each reordered sentence only contains words from at least one of the reference translations, the uni-gram precision is always 1.0. The language model probability is then computed for each reordering. Table 1 shows the average correlations between language model probabilities and BLEU scores.

We can see that the surface language model correlates moderately well with BLEU, explaining about 49% ($r^2 = 0.49$) of the variation, whereas the POS language model does not correlate with BLEU at all.[1] On the other hand, local language models alone (as introduced in Section 3) correlate with BLEU only slightly worse than surface models. The highest correlation is seen when they are interpolated with word models. The BLEU scores in Table 1 are computed using the 1-best sentences after reranking. These system-agnostic correlation results look promising for our local models and the end-to-end translation results in Section 5 confirm these initial findings.

LM	Kendall's τ	Pearson r	BLEU[%]
wordLM	0.53	0.71	80.20
POS 7gLM	0.01	0.01	48.44
locLM	0.45	0.62	76.03
λwordLM+$(1-\lambda)$locLM ($\lambda = 0.92$)	0.54	0.73	80.98

Table 1: Correlation between randomly permuted English reference translations and BLEU.

3 Local Language Models

In this section, we introduce a novel approach to language modeling that is more context-sensitive than standard POS language models. Instead of using one global POS language model that is built by using all of a mono-lingual corpus in the target language, we build individual models, or local models, for each word-POS pair using the POS tags surrounding each occurrence of that pair. This adds an aspect of lexicalization that is entirely absent in previous POS language models. The effect is that the resulting n-gram probability distributions of each local model are more biased towards the contextual constraints of each individual word-POS pair. This is similar to the idea of cached language models (Kuhn, 1988), but more fine-grained and with a tighter integration of POS and lexical information.

3.1 Definition of Local Language Models

Each conditional probability of order n in a local model for the word-POS pair $w:t$ is of the form:

$$p_{w:t}(t_n, p_n | t_1 : p_1, \ldots, t_{n-1} : p_{n-1})$$

where t_i refers to POS tags and p_i to positions relative to an occurrence of the pair $(w:t)$. For example, consider the sentence fragment in Figure 1. The conditional local n-gram probabilities (a–d) are generated from the occurrence of the word *told* with POS tag VBD. Probability (c) in Figure 1 estimates that a word with POS tag NN occurs two positions to the right of *told*, given the n-gram history that a noun occurs to its left and a determiner to its right.

[1] Interpolating both models does not lead to further correlation improvements.

position	...	11	12	13	14	15	16	17	...
relative position	...	-3	-2	-1	0	+1	+2	+3	...
word	...	the	new	mayor	told	the	reporter	to	...
POS	...	DT	JJ	NN	VBD	DT	NN	TO	...

(a) $p_{told:\mathrm{VBD}}(\mathrm{NN{:}-1}|\mathrm{DT{:}-3\ JJ{:}-2})$

(b) $p_{told:\mathrm{VBD}}(\mathrm{DT{:}+1}|\mathrm{JJ{:}-2\ NN{:}-1})$

(c) $p_{told:\mathrm{VBD}}(\mathrm{NN{:}+2}|\mathrm{NN{:}-1\ DT{:}+1})$

(d) $p_{told:\mathrm{VBD}}(\mathrm{TO{:}+3}|\mathrm{DT{:}+1\ NN{:}+2})$

Figure 1: Sentence fragment with the tri-gram probabilities (a–d) linked to *told*.

For each local model we use a sliding window considering all n-grams of length n starting n words to the left and ending n words to the right of an occurrence of the word-POS pair of the model at hand.

All local model probabilities are smoothed using Witten-Bell smoothing and interpolation.[2] POS tags are annotated with positional information to distinguish between lower-order estimates such as $p_{told:\mathrm{VBD}}(\mathrm{NN{+}2})$ and $p_{told:\mathrm{VBD}}(\mathrm{NN{+}3})$ both of which can arise when backing off during smoothing. Without positional information, $p_{told:\mathrm{VBD}}(\mathrm{NN})$ only estimates the probability of the tag NN occurring within the proximity of *told*.[3]

A local model of order n contains the conditional probabilities for words occurring at relative positions -1, +1, ...+n. Therefore the probability of a word occurrence is estimated by all local models covering this word's position. Figure 2 shows schematically how overlapping n-gram probabilities interact. E.g., the probability of word w_{i+2} is based on the probability of the local model for w_{i+1}, w_i, w_{i-1}, and w_{i-2} (the last two are not shown in Figure 2 for space reasons). Formally, the conditional probability of a word-POS pair, given its word and POS tag history is defined in Equation 5.

$$p(w_i, t_i \mid w_{i-n+1}^{i-1}, t_{i-n+1}^{i-1}) = p_{w_i:t_i}(t_{i-1}\text{:-1} \mid \langle t_{i-n}\text{:-}n, \ldots, t_{i-2}\text{:-2}\rangle) \cdot \prod_{j=0}^{n-1} p_{w_{i-n+j}:t_{i-n+j}}(t_i\text{:}n-j \mid H_{i,n}[j,\cdot]) \quad (5)$$

[2]The smaller event space of local models often leads to incomplete counts-of-counts, preventing the use of Kneser-Ney smoothing (Chen and Goodman, 1999).

[3]Despite the notational similarities, our approach should not be confused with projected POS models, which use source side POS tags to model reordering (Och et al., 2004).

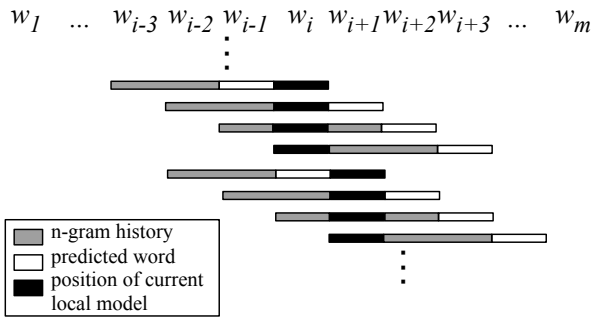


Figure 2: Schema of overlapping local language model applications.

where $H_{i,n}$ is an $n \times n$ matrix specifying the history of the word at position i. Each row j of $H_{i,n}$ represents the history of the conditional probability belonging to the local model associated with position $i-n+j$. Each entry $H_{i,n}[j,k]$ is defined as follows:

$$H_{i,n}[j,k] = \begin{cases} t_{i-n+k}\text{:}k-j & \text{if } j \neq k \\ \epsilon & \text{otherwise} \end{cases}$$

where t_{i-n+k} is the POS tag at position $i-n+k$ and $k-j$ is the relative position with respect to the diagonal of $H_{i,n}$, i.e., the position of the local language model corresponding to row j. $H_{i,n}[j,\cdot]$ is the jth row vector from which the jth entry (the empty element) has been removed. For instance, given the example in Figure 1, $H_{14,3}$ is

$$H_{14,3} = \begin{bmatrix} \epsilon & \mathrm{JJ{:}+1} & \mathrm{NN{:}+2} \\ \mathrm{DT{:}-1} & \epsilon & \mathrm{NN{:}+1} \\ \mathrm{DT{:}-2} & \mathrm{JJ{:}-1} & \epsilon \end{bmatrix}$$

For convenience we assume that the row and column indices are 0-based, i.e., the upper-left entry of a matrix is referred to by $H_{i,n}[0,0]$. In this example, $H_{14,3}[1,\cdot] = \langle \mathrm{DT{:}-1},\ \mathrm{NN{:}+1} \rangle$.

position	0	1	2	3	4	5	6
token	<s>	cuba	frees	more	dissidents	.	</s>
POS tag	<s>	NNP	VBZ	JJR	NNS	.	</s>

$$
\begin{aligned}
p(\text{cuba, NNP}|w_0^0, t_0^0) &= p_{\text{cuba:NNP}}(\text{<s>:-1}) \cdot p_{\text{<s>:<s>}}(\text{NNP:+1}) \\
p(\text{frees, VBZ}|w_0^1, t_0^1) &= p_{\text{frees:VBZ}}(\text{NNP:-1}|\text{<s>:-2}) \cdot p_{\text{<s>:<s>}}(\text{VBZ:+2}|\text{NNP:+1}) \\
&\quad \cdot p_{\text{cuba:NNP}}(\text{VBZ:+1}|\text{<s>:-1}) \\
p(\text{more, JJR}|w_0^2, t_0^2) &= p_{\text{more:JJR}}(\text{VBZ:-1}|\text{NNP:-3 VBZ:-2}) \cdot p_{\text{<s>:<s>}}(\text{JJR:+3}|\text{NNP:+1 VBZ:+2}) \\
&\quad \cdot p_{\text{cuba:NNP}}(\text{JJR:+2}|\text{<s>:-1 VBZ:+1}) \cdot p_{\text{frees:VBZ}}(\text{JJR:+1}|\text{<s>:-2 NNP:-1}) \\
p(\text{dissidents, NNS}|w_1^3, t_1^3) &= p_{\text{dissidents:NNS}}(\text{JJR:-1}|\text{NNP:-3 VBZ:-2}) \cdot p_{\text{cuba:NNP}}(\text{NNS:+3}|\text{VBZ:+1 JJR:+2}) \\
&\quad \cdot p_{\text{frees:VBZ}}(\text{NNS:+2}|\text{NNP:-1 JJR:+1}) \cdot p_{\text{more:JJR}}(\text{NNS:+1}|\text{NNP:-2 VBZ:-1}) \\
p(.\,, .|w_2^4, t_2^4) &= p_{.:.}(\text{NNS:-1}|\text{VBZ:-3 JJR:-2}) \cdot p_{\text{frees:VBZ}}(\text{.:+3}|\text{JJR:+1 NNS:+2}) \\
&\quad \cdot p_{\text{more:JJR}}(\text{.:+2}|\text{VBZ:-1 NNS:+1}) \cdot p_{\text{dissidents:NNS}}(\text{.:+1}|\text{VBZ:-2 JJR:-1}) \\
p(\text{</s>, </s>}|w_3^5, t_3^5) &= p_{\text{</s>:</s>}}(\text{.:-1}|\text{JJR:-3 NNS:-2}) \cdot p_{\text{more:JJR}}(\text{</s>:+3}|\text{NNS:+1 .:+2}) \\
&\quad \cdot p_{\text{dissidents:NNS}}(\text{</s>:+2}|\text{JJR:-1 .:+1}) \cdot p_{.:.}(\text{</s>:+1}|\text{JJR:-2 NNS:-1})
\end{aligned}
$$

Figure 3: Language model probability computation for the sentence "Cuba frees more dissidents." using our local language modeling approach.

The example in Figure 3 shows word-by-word how tri-gram local language models are used to compute the probability of a whole sentence.

Our local language model approach also bears some resemblance to statistical approaches to modeling subcategorization frames (Manning, 1993). While our approach is more general by considering all words and not just focusing on verbal subcategorization frames, it is also more shallow in the sense that only part-of-speech categories are considered which does not model any contextual relationships on the phrase level.

3.2 Building Local Language Models

To build the local language models, we use the SRILM toolkit (Stolcke, 2002), which is commonly applied in speech recognition and statistical machine translation. While SRILM collects n-gram statistics from all n-grams occurring in a corpus to build a single global language model, we build a language model for each word-POS pair only using the n-grams within the proximity of occurrences for that word-POS pair in a POS-tagged corpus. This results in separate n-gram count files, which are then processed by SRILM to build the individual language models.[4] Charniak's parser (Charniak, 2000) is used to POS tag the corpus.

[4]The pre-processing scripts are available at `http://www.science.uva.nl/~christof/locLM/`.

3.3 Decoder Integration

Several approaches that integrate POS language models have focused on n-best list re-ranking only (Hasan et al., 2006; Wang et al., 2007). Often this is due to the computational (and implementational) complexities of integrating more complex language models with the decoder, although it is expected that a tighter integration with the decoder itself leads to better improvements than n-best list re-ranking.

Integrating our local language modeling approach with a decoder is straightforward. Our baseline decoder already uses SRILM's API for computing word language model probabilities. Since SRILM supports arbitrarily many language models, local language models can be added using the same functionalities of SRILM's API. For the experiments discussed in Section 4, we add about 150,000 local language models to the word model. All local language model probabilities are coupled with the same feature weight. Potentially, improvements could be gained from using separate weights for individual local models, but this would require an optimization procedure such as MIRA (Chiang et al., 2009), which can handle a larger number of features.

During decoding no POS tagging ambiguities are resolved. Each target phrase is associated with its most likely POS tag sequence, given the source and target side of the phrase pair; see Section 2.1.

4 Experimental Setup

Three approaches are compared in our experiments: the baseline system is a phrase-based statistical machine translation system (Koehn et al., 2003), very similar to Moses (Koehn et al., 2007), using a word-based 5-gram language model. The second approach extends the baseline by including a 7-gram POS-based language model. The third approach represents the work described in this paper, extending the baseline by including 4-gram local language models.

Translation quality is evaluated for two language pairs: Arabic-to-English and Chinese-to-English. NIST's MT-Eval test sets are used for both pairs. Only resources allowed under NIST's constrained data conditions are used to train the language, translation, and lexicalized distortion models.

To see whether our local language models result in improvements over a competitive baseline, we designed the baseline to use a large 5-gram word language model and lexicalized distortion modeling, both of which are known to cancel-out improvements gained from POS language models (Birch et al., 2007; Kirchhoff and Yang, 2005). The 5-gram word language model is trained on the Xinhua and AFP sections of the Gigaword corpus (3rd edition, LDC2007T40) and the target side of the bitext. We removed from the training data all documents released during the periods that overlap with the publication dates of the documents included in our development or test data sets. In total, 630 million tokens were used to build the word language model. The language model was trained using SRILM with modified Kneser-Ney smoothing and interpolation (Chen and Goodman, 1999). It is common practice not to include higher-order n-grams that occur fewer than a predefined number of times. Here, we applied rather conservative cut-offs, by ignoring 3-, 4-, and 5-grams that occurred only once. The 7-gram POS and 4-gram local language models were both trained on the POS tagged English side of the bitext and 10M sentences from Gigaword's Xinhua and AFP sections.

The data for building the translation models were primarily drawn from the parallel news resources distributed by the Linguistic Data Consortium (LDC).[5] The Arabic-English bitext consists of 11.4M source and 12.6M target tokens, and the Chinese-English bitext of 10.6M source and 12.3M target tokens. Word alignment was performed running GIZA++ in both directions and generating the symmetric alignments using the 'grow-diag-final-and' heuristics.

All three approaches, including the baseline, use lexicalized distortion, distinguishing between monotone, swap, and discontinuous reordering, all with respect to the previous and next phrase (Koehn et al., 2005). The distortion limit is set to 5 for Arabic-to-English, and 6 for Chinese-to-English. For each source phrase the top 30 translations are considered.

For tuning and testing we use NIST's official MT-Eval test sets. MT04 was used as the development set for both language pairs. Testing was carried out on MT05 to MT09 for Arabic-English and MT05 to MT08 for Chinese-English. NIST did not release a new Chinese-English test set for MT-Eval 2009. Parameter tuning of the decoder was done with minimum error rate training (MERT) (Och, 2003), adapted to BLEU maximization.

As evaluation metrics we used NIST's adaptation of BLEU-4 (Papineni et al., 2001), version 13a, where the brevity penalty is based on the reference translation with the closest length, and translation error rate (TER) version 0.7.25 (Snover et al., 2006). All results reported here are case-insensitive. TER scores are shown as 1-TER.

To see whether the differences between the approaches we compared in our experiments are statistically significant, we apply approximate randomization (Noreen, 1989); Riezler and Maxwell (2005) have shown that approximate randomization is less sensitive to Type-I errors, i.e., less likely to falsely reject the null hypothesis, than bootstrap resampling (Koehn, 2004) in the context of machine translation.

5 Results and Analysis

The Arabic-to-English results are shown in Table 2, and the Chinese-to-English results in Table 3. All results are subdivided by genre following NIST's genre classification. Note that MT06 con-

[5] LDC catalog numbers for Arabic-English: LDC2004E72, LDC2004T17, LDC2004T18, LDC2005E46, LDC2005E83, LDC2006E25, LDC2006E34, LDC2006E85, LDC2006E92, and LDC2007T08. For Chinese-English: LDC2002E18, LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2006E34, LDC2006E85, and LDC2006E92.

systems and improvements		MT04 tune	MT05 NW	MT06 NW	MT06 WB	MT06 ALL	MT08 NW	MT08 WB	MT08 ALL	MT09 NW	MT09 WB	MT09 ALL	MT05–09 NW	MT05–09 WB	MT05–09 ALL
BLEU[%]															
1a	wordLM	51.90	53.83	46.76	34.69	43.41	48.77	33.26	42.37	52.97	34.25	44.34	50.51	34.00	45.63
2a	+posLM	51.92	54.29	47.02	34.44	43.51	48.81	33.30	42.31	53.52	34.04	44.36	50.89	33.87	45.70
3a	> wordLM	+0.02	+0.46$^{\blacktriangle}$	+0.26	−0.25	+0.10	+0.04	+0.04	−0.06	+0.55$^{\blacktriangle}$	−0.21	+0.02	+0.38$^{\blacktriangle}$	−0.13	+0.07
4a	+locLM	**52.65**	**55.08**	**47.24**	**35.17**	**43.88**	**49.61**	**33.67**	**42.92**	**54.39**	**34.40**	**44.82**	**51.57**	**34.33**	**46.22**
5a	> wordLM	+0.75$^{\blacktriangle}$	+1.25$^{\blacktriangle}$	+0.48$^{\blacktriangle}$	+0.48$^{\triangle}$	+0.47$^{\blacktriangle}$	+0.84$^{\blacktriangle}$	+0.41	+0.55$^{\blacktriangle}$	+1.42$^{\blacktriangle}$	+0.15	+0.48$^{\blacktriangle}$	+1.06$^{\blacktriangle}$	+0.33$^{\triangle}$	+0.59$^{\blacktriangle}$
6a	> +posLM	+0.73$^{\blacktriangle}$	+0.79$^{\blacktriangle}$	+0.22	+0.73$^{\blacktriangle}$	+0.37$^{\triangle}$	+0.80$^{\blacktriangle}$	+0.37	+0.61$^{\blacktriangle}$	+0.87$^{\blacktriangle}$	+0.36	+0.46$^{\blacktriangle}$	+0.68$^{\blacktriangle}$	+0.46$^{\blacktriangle}$	+0.52$^{\blacktriangle}$
1-TER[%]															
1b	wordLM	58.32	59.04	54.27	45.62	51.68	55.59	44.41	50.69	59.90	46.43	53.03	56.94	45.49	53.13
2b	+posLM	58.54	59.72	**54.90**	**45.67**	**52.14**	55.75	44.64	50.89	60.49	46.72	53.47	57.46	45.70	53.55
3b	> wordLM	+0.22$^{\triangle}$	+0.68$^{\blacktriangle}$	+0.63$^{\blacktriangle}$	+0.05	+0.46$^{\blacktriangle}$	+0.16	+0.23	+0.20$^{\triangle}$	+0.59$^{\blacktriangle}$	+0.29$^{\triangle}$	+0.44$^{\blacktriangle}$	+0.52$^{\blacktriangle}$	+0.21$^{\blacktriangle}$	+0.42$^{\blacktriangle}$
4b	+locLM	**58.95**	**60.06**	54.88	45.62	52.11	**56.42**	**44.91**	**51.38**	**60.91**	**46.84**	**53.74**	**57.79**	**45.83**	**53.81**
5b	> wordLM	+0.63$^{\blacktriangle}$	+1.02$^{\blacktriangle}$	+0.61$^{\blacktriangle}$	+0.00	+0.43$^{\blacktriangle}$	+0.83$^{\blacktriangle}$	+0.50$^{\blacktriangle}$	+0.69$^{\blacktriangle}$	+1.01$^{\blacktriangle}$	+0.41$^{\triangle}$	+0.71$^{\blacktriangle}$	+0.85$^{\blacktriangle}$	+0.34$^{\blacktriangle}$	+0.68$^{\blacktriangle}$
6b	> +posLM	+0.41$^{\blacktriangle}$	+0.34$^{\triangle}$	−0.02	−0.05	−0.03	+0.67$^{\blacktriangle}$	+0.27	+0.49$^{\blacktriangle}$	+0.42$^{\triangle}$	+0.12	+0.27$^{\triangle}$	+0.33$^{\blacktriangle}$	+0.13	+0.26$^{\blacktriangle}$
# segments		1,353	1,056	1,033	764	1,797	813	547	1,360	586	727	1,313	3,488	2,038	5,526

Table 2: Results for Arabic-to-English translation. Comparison of our approach (**+locLM**, rows 4a/b) to the baseline using a word language model (**wordLM**, rows 1a/b) and a competing approach using a POS-based language model (**+posLM**, rows 2a/b). Results are presented using BLEU[%] (rows 1a–6a) and 1-TER[%] (rows 1b–6b) and broken down by genre: NW=newswire, WB=web, and ALL=NW∪WB. Rows 3a/b, 5a/b, and 6a/b show the relative improvements over the system mentioned to the right of the $>$ sign. Statistically significant improvements/declines (using approximate randomization) at the $p < .01$ level are marked $^{\blacktriangle}$/$^{\blacktriangledown}$ and $^{\triangle}$/$^{\triangledown}$ at the $p < .05$ level.

tains the genres 'broadcast news' and 'newsgroup'. In both tables, the former has been classified under 'newswire' and the latter under 'web'.

The first approach is the baseline system 'wordLM' (rows 1a/b in Tables 2 and 3), which uses a 5-gram word-based language model. The next approach '+posLM' extends the baseline by adding a 7-gram POS language model (rows 2a/b in both tables). Rows 3a/b show the relative improvements over the baseline. The third approach '+locLM' (rows 4a/b) uses local language models in addition to the baseline's word-based model. Note that +locLM does not use the 7-gram POS language model as well. Rows 5a/b show the relative improvements of the local modeling approach over the baseline and rows 6a/b the improvements over the approach using a POS language model.

Let us first take a closer look at the Arabic-to-English results in Table 2. The approach using a POS language model results in statistically significant improvements for only one test set (MT05) and the newswire documents of MT09. The average improvements across all sets and genres are negligible ($+0.07$ BLEU). Our local language modeling approach achieves the highest BLEU scores for all test sets and across all genres. In particular, the improvements of +1.06 BLEU for newswire documents are substantial. With the exception of MT08-WB and MT09-WB all BLEU improvements over the baseline are statistically significant.

When evaluating with 1-TER, local language modeling also achieves the best results, with the exception of MT06, where the POS language model approach performs slightly better.

Turning to the Chinese-English results in Table 3, we see similar improvements in BLEU. The improvements of using a POS language model are negligible ($+0.04$ BLEU). Here as well, local language modeling leads to the best results, with substantial improvements of +0.88 BLEU for web documents.

The major difference between Arabic-English and Chinese-English is the discrepancy between BLEU score improvements and decreases in 1-TER. While we cannot explain this discrepancy, it is worth noting that similar discrepancies between BLEU and TER and Arabic-to-English and Chinese-to-English translation can be found in the literature. The results described in Shen et al. (2009) show a strong correlation between BLEU and 1-TER improvements[6]

[6]Shen et al. (2009) report TER rather than 1-TER scores.

systems and improvements		MT04 tune	MT05 NW	MT06 NW	MT06 WB	MT06 ALL	MT08 NW	MT08 WB	MT08 ALL	MT05–08 NW	MT05–08 WB	MT05–08 ALL
BLEU[%]												
1a	wordLM	37.32	32.55	33.33	23.40	31.16	28.67	17.57	24.03	31.93	19.82	29.30
2a	+posLM	37.32	32.47	33.13	23.67	31.06	28.63	**18.46**	24.35	31.82	20.46	29.34
3a	> wordLM	+0.00	−0.08	−0.20	+0.27	−0.10	−0.04	+0.89$^{\blacktriangle}$	+0.32	−0.11	+0.64$^{\blacktriangle}$	+0.04
4a	+locLM	**38.15**	**33.05**	**33.33**	**24.62**	**31.42**	**29.52**	18.24	**24.79**	**32.36**	**20.70**	**29.82**
5a	> wordLM	+0.83$^{\blacktriangle}$	+0.50$^{\triangle}$	+0.00	+1.22$^{\blacktriangle}$	+0.26	+0.85$^{\blacktriangle}$	+0.67$^{\triangle}$	+0.76$^{\blacktriangle}$	+0.43$^{\blacktriangle}$	+0.88$^{\blacktriangle}$	+0.52$^{\blacktriangle}$
6a	> +posLM	+0.83$^{\blacktriangle}$	+0.58$^{\blacktriangle}$	+0.20	+0.95$^{\blacktriangle}$	+0.36$^{\triangle}$	+0.89$^{\blacktriangle}$	−0.22	+0.44$^{\triangle}$	+0.54$^{\blacktriangle}$	+0.24	+0.48$^{\blacktriangle}$
1-TER[%]												
1b	wordLM	**42.81**	**40.73**	**42.99**	**39.42**	**42.15**	**40.42**	**36.77**	**38.78**	**41.53**	**37.77**	**40.63**
2b	+posLM	42.50	40.60	42.75	38.87	41.84	39.76	36.75	38.41	41.23	37.55	40.34
3b	> wordLM	−0.31$^{\triangledown}$	−0.13	−0.24	−0.55	−0.31$^{\triangledown}$	−0.66$^{\blacktriangledown}$	−0.02	−0.37$^{\triangledown}$	−0.30$^{\blacktriangledown}$	−0.22	−0.29$^{\blacktriangledown}$
4b	+locLM	42.77	40.49	42.62	39.40	41.86	40.00	36.11	38.26	41.20	37.35	40.27
5b	> wordLM	−0.04	−0.24	−0.37	−0.02	−0.29	−0.42	−0.66$^{\blacktriangledown}$	−0.52$^{\blacktriangledown}$	−0.33$^{\triangledown}$	−0.42$^{\triangledown}$	−0.36$^{\blacktriangledown}$
6b	> posLM	+0.27	−0.11	−0.13	+0.53	+0.02	+0.24	−0.64$^{\blacktriangledown}$	−0.15	−0.03	−0.20	−0.07
# segments		1,788	1,082	1,181	483	1,664	691	666	1,357	2,954	1,149	4,103

Table 3: Comparison of our system for Chinese-to-English translation. See Table 2 for details on notation.

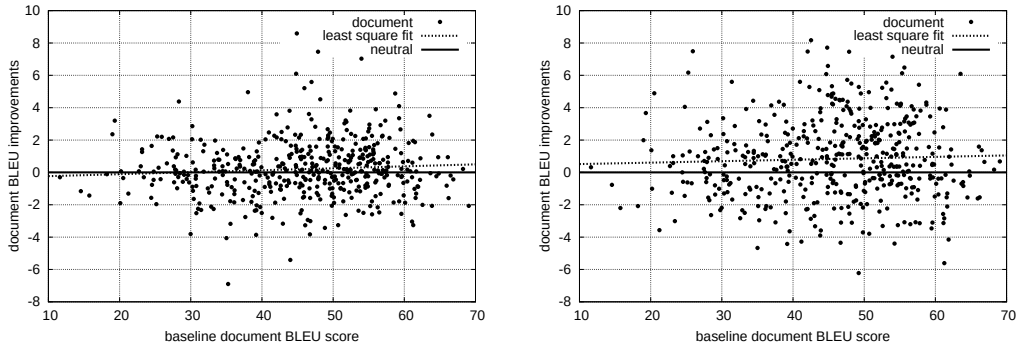
for Arabic-to-English on the MT06 and MT08 sets, but for Chinese-to-English the correlation seems to be much weaker and BLEU improvements of +0.75 can correspond to decreases of up to -0.80 in 1-TER.

One of the motivations of using POS language models in general, and local language models in our case, is to improve the fluency of translations, which should be reflected in increased precision for higher-order n-grams. Table 4 shows that this is the case when comparing local modeling to both word and POS language models for Arabic-to-English translation. The same trend, but to a somewhat weaker degree can be observed for Chinese-to-English.

	Prec-1	Prec-2	Prec-3	Prec-4	BP
Arabic-English (MT05–09)					
wordLM	81.38	54.51	38.10	26.99	0.987
+posLM	81.81	54.82	38.34	27.17	0.983
+locLM	81.90	55.35	39.01	27.86	0.981
Chinese-English (MT05–08)					
wordLM	75.03	40.56	22.55	12.93	0.955
+posLM	74.81	40.30	22.41	12.83	0.962
+locLM	74.24	40.70	22.83	13.19	0.966

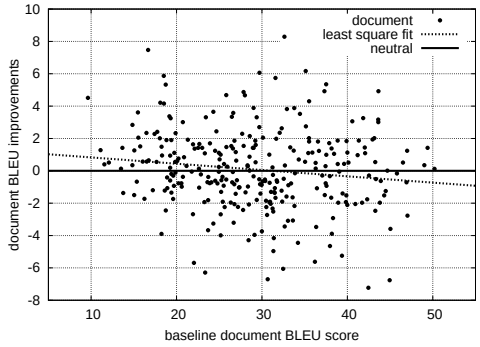
Table 4: BLEU n-gram precision ($1 \leq n \leq 4$) and Brevity Penalty (BP) scores over all test sets.

The effectiveness of a POS language model often diminishes with improved translation quality of the base system to which it is added. Naturally, we are interested in the extent that this diminishing effect also holds for our local language models. A full experimental setup, varying all relevant factors, such as language, translation, and distortion model size, and the various meta-parameters, is beyond the scope of this paper. Nevertheless, we can gauge this by taking a closer look at the distribution of improvements within our experiments. Figure 4 shows performance improvements in document-level BLEU for both language pairs. The document-level BLEU score for the baseline system is plotted on the x-axis and improvements are plotted on the y-axis. The dotted line is the linear fit (using least square regression). If the effectiveness of either added model (POS or local) diminishes with increasing translation quality, we would expect a declining regression line. This is not the case for Arabic-to-English translation. Relative improvements for both added models increase as the translation quality of the baseline increases. The slope of both regression fits is almost identical, but the y-intercept is larger for our local modeling approach. Note that the small slope is also due to difference in scale between full BLEU scores and relative improvements. We can observe the opposite for Chinese-to-English translation, where the slope is negative. Both models seem to help more for documents with lower baseline translation quality. For the POS model, the regression line intersects with the neutral line (± 0 improvement) at around 31 BLEU, which is close to the average BLEU score and in line with its negligible improvements (see Ta-

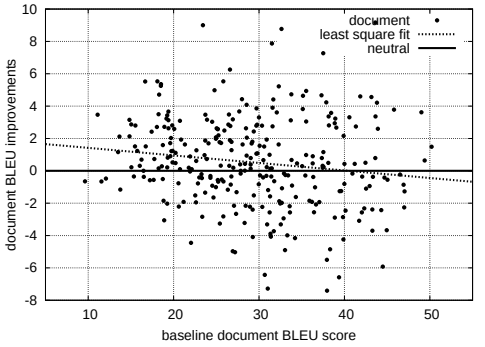


Arabic-English: +posLM > wordLM

Arabic-English: +locLM > wordLM



Chinese-English: +posLM > wordLM



Chinese-English: +locLM > wordLM

Figure 4: Correlation between baseline BLEU scores for individual documents and the relative, absolute improvements achieved by +posLM (left) and +locLM (right). BLEU scores (and improvements) are computed at the document level.

ble 3). For the local language model, the regression line intersects with the neutral line at about 40 BLEU, suggesting that until translation quality improves substantially, local language models could still have a positive impact.

6 Related Work

The main goal of this paper is to show that by tying POS language models to lexical items, we get more accurate distributions for specific words. The work on factored language models (Bilmes and Kirchhoff, 2003) is related to our work to the extent that it also mixes POS tags with lexical information, albeit in a very different manner. Factored language models use more general representations, such as POS tags or stems, only during back-off. Kirchhoff and Yang (2005) applied factored language models to machine translation but the improvements were negligible.

Collins et al. (2005) proposed a discriminative language modeling approach that uses mixtures of POS and surface information and showed that it leads to a reduction in speech recognition word error rates. On the other hand, their approach seems more suited for n-best list re-ranking and it is not clear whether those improvements carry over to machine translation. Li and Khudanpur (2008) adapted this discriminative approach to machine translation re-ranking but used surface forms only.

Wang et al. (2007) and Zheng et al. (2008) use elaborately enriched representations, called *super abstract role values* (Wang and Harper, 2002), which capture contextual dependencies using lexical categories, role labels, and dependency grammar structures. So far their approach has been limited to re-ranking n-best lists only.

7 Conclusion

Though POS language models do not lead to significant improvements over a competitive baseline, we have shown that a competitive phrase-based baseline system can benefit from using POS information by building lexically anchored local models. Our local model approach does not only lead to more context-specific probability distributions, but also takes ad-

vantage of the language model probability of each word being based on all surrounding local models. The evaluations for Arabic- and Chinese-to-English show that local models lead to statistically significant improvements across different test sets and genres. Correlating the translation quality of the baseline with the improvements that result from adding local models, further suggests that these improvements are sustainable and should carry over to improved baseline systems.

Acknowledgments

This research was funded in part by the European Commission through the CoSyne project FP7-ICT-4-248531, the European Commission's ICT Policy Support Program as part of the Competitiveness and Innovation Framework Program, CIP ICT-PSP under grant agreement nr. 250430, and the PROMISE Network of Excellence co-funded by the 7th Framework Programme of the European Commission, grant agreement no. 258191.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 529–536.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 4–6.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16.

Hélène Bonneau-Maynard, Alexandre Allauzen, Daniel Déchelotte, and Holger Schwenk. 2007. Combining morphosyntactic enriched representation with n-best reranking in statistical translation. In *Proceedings of the NAACL-HLT Workshop on Syntax and Structure in Statistical Translation*, pages 65–71.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–393.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 218–226.

Michael Collins, Brian Roark, and Murat Saraclar. 2005. Discriminative syntactic language modeling for speech recognition. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 507–514.

Saša Hasan, Oliver Bender, and Hermann Ney. 2006. Reranking translation hypotheses using structural properties. In *Proceedings of the EACL Workshop on Learning Structured Information in Natural Language Applications*, pages 41–48.

Peter Heeman. 1998. POS tagging versus classes in language modeling. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 179–187.

Katrin Kirchhoff and Mei Yang. 2005. Improved language modeling for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 125–128.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Inter-*

active Poster and Demonstration Sessions, pages 177–180.

Philipp Koehn, Abhishek Arun, and Hieu Hoang. 2008. Towards better machine translation quality for the german–english language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 139–142.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Roland Kuhn. 1988. Speech recognition and the frequency of recently used words: a modified Markov model for natural language. In *Proceedings of the 12th conference on Computational Linguistics*, pages 348–350.

Zhifei Li and Sanjeev Khudanpur. 2008. Large-scale discriminative n-gram language models for statisti- cal machine translation. In *Proceedings of AMTA*, pages 133–142.

Christopher D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 235–242.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19:313–330.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley-Interscience.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, ALex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the 2004 Meeting of the North American chapter of the Association for Computational Linguistics*, pages 161–168.

Franz-Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2001)*, pages 311–318.

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 172–181.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21:492–518.

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 72–80.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Andreas Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL-04)*, pages 101–104.

Wen Wang and Mary P. Harper. 2002. The SuperARV language model: investigating the effectiveness of tightly integrating multiple knowledge sources. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 238–247.

Wen Wang, Andreas Stolcke, and Jing Zheng. 2007. Reranking machine translation hypotheses with structured and web-based language models. In *IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 159–164.

Jing Zheng, Necip Fazil Ayan, Wen Wang, Dimitra Vergyri, Nicolas Scheffer, and Andreas Stolcke. 2008. SRI systems in the NIST MT08 Evaluation. In *Proceedings of the NIST 2008 Open MT Evaluation Workshop*.