Statistical Machine Translation as a Language Model for Handwriting Recognition

Jacob Devlin, Matin Kamali, Krishna Subramanian, Rohit Prasad and Prem Natarajan
Raytheon BBN Technologies
10 Moulton St, Cambridge, MA, 02138
{*jdevlin, mkamali, ksubrama, rprasad, prem*}*@bbn.com*

Abstract

When performing handwriting recognition on natural language text, the use of a word-level language model (LM) is known to significantly improve recognition accuracy. The most common type of language model, the n-gram model, decomposes sentences into short, overlapping chunks.

In this paper, we propose a new type of language model which we use in addition to the standard n-gram LM. Our new model uses the likelihood score from a statistical machine translation system as a reranking feature. In general terms, we automatically translate each OCR hypothesis into another language, and then create a feature score based on how "difficult" it was to perform the translation. Intuitively, the difficulty of translation correlates with how well-formed the input sentence is. In an Arabic handwriting recognition task, we were able to obtain an 0.4% absolute improvement to word error rate (WER) on top of a powerful 5-gram LM.

1 Introduction

Language modeling is a crucial component in a number of natural language processing tasks, including Optical Character Recognition (OCR). The purpose of language modeling is to determine how "valid" a natural language sentence is, independent of the input signal which was used to generate the sentence. Modern OCR systems primarily use an *n-gram language model* (LM), which treats each sentence as a series of short, overlapping chunks, called n-grams.

In this paper, we propose a new types of language model, which is based on the translation likelihood of a statistical machine translation (SMT) system. To give an example, imagine that we have an Arabic OCR hypothesis sentence for which we would like to compute the language model score. We simply feed this sentence into an Arabic-to-English (or Arabic-to-Anything) SMT system, and generate an English translation of the Arabic sentence. Then, we can use the translation likelihood score produced by the SMT system as an additional feature in our OCR system, while the English translation itself is discarded. We refer to this model as the *SMT-LM*.

To give intuitive support to our approach, one can imagine that at a high level, a "better" input sentence will be "easier" for the SMT system to translate, and thus its translation will receive a higher SMT likelihood score. However, we believe that the SMT-LM also provides more principled benefits, such implicit paraphrasing, improved probability estimation, and long-distance modeling.

We follow an *k-best reranking* approach in this work. In other words, the OCR system produces a ranked list of hypotheses for each input sentence, where each hypothesis is associated with a log-likelihood score from the recognition model. Then, we use our SMT-LM as a *reranking feature*, where we obtain an SMT-LM score for each hypothesis, and re-rank each k-best list based on a linear combination of this new score and the original model score. The relative weights for each of these scores is discriminatively optimized to minimize OCR word error rate (WER).

We test our approach on an Arabic OCR task. We demonstrate that the SMT-LM provides a significant improvement in WER on top of a powerful 5-gram reranking LM, in spite of the fact there is a significant domain mismatch between the OCR system and the SMT system. Additionally, we show that the SMT-LM significantly outperforms a simple syntactic language

[1] This paper is based upon work supported by the DARPA MADCAT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.273

model, which is comparable in terms of implementation.

The paper is organized as follows. In Section 2, we describe related work. In Section 3, we give an overview of modern statistical machine translation systems, and also briefly describe the OCR system used in this paper. In Section 5, we give the intuition and implementation of using an SMT system as a reranking language model. In Section 6, we present experimental results on an OCR task. In Section 7, we describe our conclusions and future work.

2 Related Work

Our SMT-LM is in largely analogous to the *Parse-LM* approach to syntactic language modeling, where each hypothesis is fed into a statistical syntactic parser, and the parse-likelihood score is used as an additional feature in reranking [12].

Although more effective forms of syntactic language modeling have been developed [2], we feel that the simple Parse-LM provides the fairest comparison to our SMT-LM in terms of both ease and method of implementation. Both the SMT-LM and Parse-LM use the likelihood score from an independent statistical NLP system as a reranking language model. Moreover, in both cases the SMT/parse system is as a "black-box," where the k-best hypotheses are fed into the system and only the final likelihood score is used.

We know of no previous attempts to use an SMT system itself as a language model for an another NLP task. Work such as [5] has attempted to translate the output of an SMT system back into the original language, but this was generally done as an outdated method of accuracy evaluation.

3 Background

In this section we will give a brief description of the OCR system used in this paper, as well as a general overview of n-gram language modeling.

3.1 OCR System

We use an HMM-driven OCR system described in [8, 10]. The system can be divided into two basic functional components: training and recognition. Both training and recognition share a common preprocessing and feature extraction stage in which we first de-skew the scanned text zone and then locate the regions (bounding boxes) of individual text lines.

The feature extraction process computes a feature vector as a function of the horizontal position within each of these line regions. Each line of text is horizontally segmented into a sequence of thin, overlapping, vertical strips called frames. For each frame we compute a script-independent, feature vector that is a numerical representation of the frame. This process can be thought of as scanning the line of text from left to right with a fixed-width window and measuring observations from that window at regular intervals.

Our OCR system models each character with a multi-state, left-to-right HMM. Each state has an associated output probability distribution over the features. The number of states and the allowable transitions are system parameters. For Arabic, we use 14-state, left-to-right HMMs with skip states.

HMM training is performed using the Baum-Welch or Forward-Backward algorithm, which aligns the feature vectors with the character-models to obtain maximum likelihood estimates of HMM parameters. For our system the HMM parameters are the means and variances of the component Gaussians in the Gaussian mixture model of the state output probabilities, the mixture component weights and the state transition probabilities.

During recognition we search for the sequence of words that is most likely given the feature-vector sequence and the trained character-models, in accordance with the constraints imposed by a lexicon and/or language model. We use a word-based n-gram language model which is estimated from a large text corpus.

The HMM system shown in Figure 4 operates at the sentence level. We perform automatic line finding and then concatenate/split the lines so that they correspond to natural language sentences. For each sentence a ranked set of hypotheses is generated by the OCR system.

3.2 n-gram Language Modeling

The n-gram language model makes two important independence assumptions. First, it assumes the probability of a whole sentence W is computed by multiplying independent estimates for each word w in W. Second, it assumes the probability of a word w only depends on the preceding $n-1$ words. Formally, the probability of a sentence W is defined as:

$$\begin{aligned} P(W) &= \prod_i P(w_i|W) && (1) \\ &= \prod_i P(w_i|w_{i-1}, w_{i-2}, ...w_{i-n+1}) && (2) \end{aligned}$$

The probabilities are estimated using maximum likelihood estimates (MLEs) computed from a large monolingual corpus. The most basic models use unsmoothed

MLEs:

$$\begin{aligned} h &= w_{i-1}, w_{i-2}, ... w_{i-n+1} \quad (3) \\ P(w_i|h) &= \frac{C(h, w_i)}{\sum_{w'} C(h, w')} \quad (4) \end{aligned}$$

where $C(\vec{w})$ is the count of n-gram $\vec{w}$ in the corpus.

More advanced n-gram language models use "back-off" techniques to avoid data sparsity issues [4, 14] (e.g., to avoid assigning a probability of 0 to unseen n-grams).

4 Statistical Machine Translation Overview

The task of statistical machine translation (SMT) is to translate a *source* language sentence S into a *target* language sentence T using a set of probabilistic *translation rules*. The primary training data necessary to build an SMT system is a set of bilingual sentence pairs from which these translation rules are automatically extracted.

This section will only provide a brief overview of modern SMT, please see [9] for a more complete description.

4.1 Rule Extraction

To give an example, imagine that our source language is Spanish and our target language is English, and our training corpus contains the following bilingual sentence pair:[1]

```
Source:  el coche rojo es bonito
Target:  the red car is pretty
```

The exact scope of the rule inference procedure is outside the scope of this paper, but the SMT system will automatically extract the following rules, among others:

- `el` → `the`
- `el coche rojo` → `the red car`
- `es bonito` → `is pretty`

The rules are assigned conditional maximum likelihood probabilities. For example, let's say `es bonito` is seen 5 times in the training, where it is translated to `is pretty` 3 times and `is nice` 2 times. In this case, the *forward* – that is to say, "target given source" – translation probabilities are:

- P_{fw}(`is pretty | es bonito`) = `0.6`
- P_{fw}(`is nice | es bonito`) = `0.4`

The rules are also assigned *backwards* probabilities, e.g. P_{bw}(`es bonito | is pretty`), estimated in the same way.

We can then translate a *new* sentence by applying rules learned from different sentences in our bilingual training corpus:

Source: `un coche red es rapido`

Rules:

- `un` → `a`
- `coche rojo` → `red car`
- `es rapido` → `is fast`

Output: `a red car is fast`

4.2 Modeling Translation

In statistical systems, there are many different ways of translating a single input sentence, depending on which rules are applied. Each of these possible *translation hypotheses* is associated with a probability according to the translation model and language model. The hypothesis with the highest probability, known as the *Viterbi* hypothesis, is selected as the output of the system:

$$T^* = \text{argmax}_T P(T|S) \quad (5)$$

where T is a target hypothesis and S is the source input sentence.

In practice, we implement $P(T|S)$ using a log-linear feature-based model. We define the *decoding score* $D(S,T)$ as a weighted sum of log-feature-probabilities:

$$D(S,T) = \sum_i w_i * F_i(S,T) \quad (6)$$

Where F is a set of SMT features, and w_i is the weight associated with feature i. The most important features used in an SMT system are:

- $\log(P_{fw}(T|S))$ = Forward translation probability
- $\log(P_{bw}(S|T))$ = Backward translation probability
- $\log(P_{lm}(T))$ = Language model probability of T
- $|T|$ = Number of words in hypothesis T

The translation probability of a hypothesis is simply computed as the product of that hypothesis' rule probabilities.

We now define the Viterbi hypothesis the highest decoding score:

$$T^* = \text{argmax}_T D(S,T) \quad (7)$$

This search is performed using a beam search heuristic, or something similar.

[1] Spanish is used in these examples for clarify, but the actual work on this paper uses Arabic as the source language.

The SMT system used in this paper is based on [13] and [1]. This type of SMT is known a "hierarchical" system, which is contrasted with "phrasal" and "syntactic" SMT systems. However, we do *not* believe that it is crucial to use any one particular type of SMT system for the work described here.

5 SMT as a Language Model

In this section, we first show our very simple method for using an SMT system to create a reranking LM to be used in some arbitrary underlying NLP task. Next, we present our intuitions behind why we believe SMT works well as a language model. Finally, we will briefly describe the caveats of building the necessary SMT system to perform language modeling.

5.1 Implementation

If an existing SMT system is available, implementation is trivial.[2] At the OCR reranking stage, simply feed each hypothesis into the SMT system, and use the resulting Viterbi likelihood score $D(S, T^*)$ as an additional feature. This score is easily accessible in virtually every open source SMT implementation. Note that it is *theoretically* preferable to use the sum of probabilities over *all* possible translations:

$$\log \sum_{T} e^{D(S,T)} \tag{8}$$

However, in practice, SMT systems use significant pruning throughout the search, so summing over multiple hypotheses would result in an arbitrary value over a small fraction of the search space. We believe that the Viterbi score is a sensible approximation.

5.2 Intuition

At a very high level, we can describe the intuitions behind the SMT-LM using humans as an analog. We can equate standard n-gram language modeling to asking a human to classify how "valid" a particular sentence is in terms of grammatical structure, semantic meaning, etc. Of course, we know that n-gram models are a very rough and noisy approximation of this.

Next, we ask a bilingual speaker to translate each sentence, and rate how difficult it was to perform this translation. Here, we would expect that the less valid the Arabic input sentence, the more difficult it is to translate. This is represented by the SMT probabilities $P_{fw}(T|S)$ and $P_{bw}(S|T)$.

Finally, we ask an English speaker to rate how grammatical each English translation is. Again, we would expect that less valid Arabic input to produce less valid English output. This is an analog to our SMT system's internal language model, $P_{lm}(T)$.

Although this high-level explanation is presents a nice intuitive overview, we also believe that there are more principled benefits from the SMT-LM over the standard n-gram LM.

First, the SMT-LM provides implicit "paraphrasing" to mitigate the effect of arbitrary variability in natural language. In other words, there are often *many* different ways to write a semantically equivalent sentence, e.g., "The man is friendly." vs. "The man is affable.". Even though these sentences mean virtually the same thing, "friendly" is a much more common word than "affable," so we would expect the fist sentence to have a higher n-gram probability than the second.

Even though the SMT system is certainly not guaranteed to produce the same Viterbi translation for two semantically equivalent input sentences, it *does* search over all possible translations and choose one with the highest likelihood. Therefore, we would *not* expect a rare word (which has a non-rare synonym) like "affable" to be translated to an equally rare word in the target, since the target LM would heavily penalize this. Thus, we would expect that the *translations* of "The man is friendly" and "The man is affable" to have much more comparable n-gram LM scores than the input sentences themselves.

Second, the SMT system's target n-gram LM may be better estimated than the n-gram LM of the source language, simply because more data is available. For example, in many domains, a far larger amount of English data is available compared to other languages. In our case, our SMT system's LM is estimated on billions of words of English text, although our Arabic 5-gram LM itself is estimated on over 500 million words of text.

Finally, the SMT-LM perform implicit long-distance modeling due to word order difference between the two languages. For example, Arabic is a Verb-Subject-Object language, so the verb and object may be a long distance from one another. English is a Subject-Verb-Object language, so the verb and object are generally closer to one another. Therefore, it verb-object agreement may be better modeled by the SMT system's English n-gram language model than the original Arabic n-gram LM.

[2] Of course, the source language of the SMT system must match the language of the task.

5.3 Building the SMT System

If an SMT system is not available, building one can require some effort. A number of robust, open-source software packages do exist for building state-of-the-art SMT systems, including Joshua [7] and Moses [6].

Perhaps the biggest challenge in building an SMT system is obtaining suitable bilingual training data. Bilingual training data is available for a large number of languages, although we do not yet know how the domain of the data and the target language affect the capability of the SLM-LM. Popular free sources of bilingual training include the Europarl corpus [5] and the United Nations corpus [11]. The Linguistic Data Consortium (LDC)[3] provides a wide array of bilingual corpora to paying members.

In this paper, we use an Arabic-to-English SMT system due to the large amount of high-quality Arabic-to-English training available. It would also be possible to build multiple SMT-LM systems with different target languages, and use each of these as independent features. We plan to explore this in future work.

6 Experiments

In this section, we present results on an OCR k-best reranking task. Our goal is to minimize the Word Error Rate (WER) of the OCR system.

The OCR data used for this task is a large collection of Arabic documents collected from the field (legal filings, etc.). This corpus contains a mix of handwritten and machine-printed data, but we only test on the handwritten subset. The OCR training consists of 2.3 million words of transcribed text.

Table 1 shows the sizes of our development and test set. The development set here is used to optimize the weights in the baseline OCR system and in re-ranking.

	Num Sents	Num Words
Dev	1,478	23,470
Test	1,397	22,261

Table 1. Sizes of the OCR dev and test sets.

The SMT system used here is trained on 45 million words of Arabic-to-English parallel training, produced by the Linguistic Data Consortium (LDC). The English n-gram LM used by the SMT system contains 5 billion words of training from the English GigaWord corpus, also produced by LDC. The SMT system is tuned on a newswire development set.

Although our SMT system was trained on quite a large amount of data, it is important to note that this system was trained, tuned, and tested on *news* data, which is a significant mismatch from our OCR system. Therefore, the SMT system effectively acts as a black-box, off-the-shelf translator, and it was not tailored towards the OCR data in any way.

Additionally, even though runtime was not a major concern in this project, we should note that our SMT system runs at over 10,000 words per minute, so the SMT-LM reranking procedure is significantly faster than the initial recognition.

We compare our SMT-LM to two other reranking features:

- **5-gram LM** - A strong Arabic 5-gram LM, estimated on three corpora: The in-domain transcripts used to train the OCR system (2.3m words), the Arabic side of the SMT training (45m words), and the Arabic GigaWord corpus (500m words). The interpolation weights were estimated to minimize perplexity on the OCR dev set references.
- **Parse-LM** - The likelihood score from the Stanford Parser's PCFG model [3], which is trained on the Penn Arabic Treebank.

6.1 Results

Reranking Method	WER
No Reranking	24.88
Parse-LM	24.89
5-gram LM	24.21
SMT-LM	24.25
SMT-LM + 5-gram LM	23.81

Table 2. Test set WER using the SMT-LM feature for OCR 20-best reranking

In Table 2, we show the results for 20-best reranking. We can see that the SMT-LM produces an improvement of 0.4 WER on top of the 5-gram LM. This demonstrates that the SMT-LM and standard n-gram LM have a significant amount of complementary information, as we suggested in Section 5.2.

It is also interesting to note that even on its own, the SMT-LM produces roughly the same 0.6∼0.7 WER improvement as the 5-gram. Of course, the 5-gram LM is far easier to implement than the SMT-LM, so we are not

[3] http://www.ldc.upenn.edu/

suggesting that the SMT-LM is preferable to a 5-gram on its own.

However, for certain languages and domains, could potentially be easier to obtain a small amount of bilingual training data than to obtain a large amount of source data. If the target side of the bilingual data is a high-density language such as English, then the SMT-LM is *effectively* increasing the amount of LM data by several orders of magnitude.

The Parse-LM provides no benefits, even on its own. Results of the Parse-LM in the literature have been somewhat negative, so this result is not entirely surprising.

7 Conclusions and Future Work

In this paper, we explored idea of using a statistical machine translation system as a language model for an OCR handwriting recognition task. After explaining the intuition behind this approach, we showed how an out-of-the-box SMT system can be used to create a very straightforward k-best re-ranking feature to improve OCR word error rate.

We were able to obtain an improvement of 0.4 WER over a strong 5-gram reranking LM using this new feature, even though there was a significant mismatch between the domain used to train the SMT system and the domain of the OCR task. Comparatively, we obtained no gain from the more well-known method of using a syntactic parser for language modeling.

In the future, we would like to use deeper information from the SMT system rather than use it as a "blackbox."

Additionally, we plan to explore the idea of using multiple target languages as additional SMT-LM features. Finally, we plan to extend this work to other domains, such as speech recognition and SMT itself.

References

[1] D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007.

[2] M. Collins, B. Roark, and M. Saraclar. Discriminative syntactic language modeling for speech recognition. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 507–514, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[3] S. Green and C. D. Manning. Better arabic parsing: baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 394–402, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[4] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181 –184 vol.1, may 1995.

[5] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT.

[6] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[7] Z. Li, C. Callison-Burch, C. Dyer, S. Khudanpur, L. Schwartz, W. Thornton, J. Weese, and O. Zaidan. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March 2009. Association for Computational Linguistics.

[8] E. MacRostie, R. Prasad, S. Rawls, M. Kamali, H. Cao, K. Subramanian, and P. Natarajan. The bbn document analysis service: a platform for multilingual document translation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 447–454, New York, NY, USA, 2010. ACM.

[9] F. J. Och and H. Ney. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30:417–449, Dec. 2004.

[10] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan. Improvements in hidden markov model based arabic ocr. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, dec. 2008.

[11] A. Rafalovitch and R. Dale. United nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of the MT Summit XII*, pages 292–299, Ottawa, Canada, 2009.

[12] B. Roark. Probabilistic top-down parsing and language modeling. *Comput. Linguist.*, 27:249–276, June 2001.

[13] L. Shen, J. Xu, and R. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *ACL-HLT*, pages 577–585, 2008.

[14] I. Witten and T. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, 37(4):1085 –1094, jul 1991.