How Many Bits Are Needed To Store Probabilities for Phrase-Based Translation?

Marcello Federico and Nicola Bertoldi
ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
38050 Povo - Trento, Italy
{federico,bertoldi}@itc.it

Abstract

State of the art in statistical machine translation is currently represented by phrase-based models, which typically incorporate a large number of probabilities of phrase-pairs and word n-grams. In this work, we investigate data compression methods for efficiently encoding n-gram and phrase-pair probabilities, that are usually encoded in 32-bit floating point numbers. We measured the impact of compression on translation quality through a phrase-based decoder trained on two distinct tasks: the translation of European Parliament speeches from Spanish to English, and the translation of news agencies from Chinese to English. We show that with a very simple quantization scheme all probabilities can be encoded in just 4 bits with a relative loss in BLEU score on the two tasks by 1.0% and 1.6%, respectively.

1 Introduction

In several natural language processing tasks, such as automatic speech recognition and machine translation, state-of-the-art systems rely on the statistical approach.

Statistical machine translation (SMT) is based on parametric models incorporating a large number of observations and probabilities estimated from monolingual and parallel texts. The current state of the art is represented by the so-called phrase-based translation approach (Och and Ney, 2004; Koehn et al., 2003). Its core components are a translation model that contains probabilities of phrase-pairs, and a language model that incorporates probabilities of word n-grams.

Due to the intrinsic data-sparseness of language corpora, the set of observations increases almost linearly with the size of the training data. Hence, to efficiently store observations and probabilities in a computer memory the following approaches can be tackled: designing compact data-structures, pruning rare or unreliable observations, and applying data compression.

In this paper we only focus on the last approach. We investigate two different quantization methods to encode probabilities and analyze their impact on translation performance. In particular, we address the following questions:

- How does probability quantization impact on the components of the translation system, namely the language model and the translation model?
- Which is the optimal trade-off between data compression and translation performance?
- How do quantized models perform under different data-sparseness conditions?
- Is the impact of quantization consistent across different translation tasks?

Experiments were performed with our phrase-based SMT system (Federico and Bertoldi, 2005) on two large-vocabulary tasks: the translation of European Parliament Plenary Sessions from Spanish to

Proceedings of the Workshop on Statistical Machine Translation, pages 94–101,
New York City, June 2006. ©2006 Association for Computational Linguistics

English, and the translation of news agencies from Chinese to English, according to the set up defined by the 2005 NIST MT Evaluation Workshop.

The paper is organized as follows. Section 2 reviews previous work addressing efficiency in speech recognition and information retrieval. Section 3 introduces the two quantization methods considered in this paper, namely the Lloyd's algorithm and the Binning method. Section 4 briefly describes our phrase-based SMT system. Sections 5 reports and discusses experimental results addressing the questions in the introduction. Finally, Section 6 draws some conclusions.

2 Previous work

Most related work can be found in the area of speech recognition, where n-gram language models have been used for a while.

Efforts targeting efficiency have been mainly focused on pruning techniques (Seymore and Rosenfeld, 1996; Gao and Zhang, 2002), which permit to significantly reduce the amount of n-grams to be stored at a negligible cost in performance. Moreover, very compact data-structures for storing back-off n-gram models have been recently proposed by Raj and Whittaker (2003).

Whittaker and Raj (2001) discuss probability encoding as a means to reduce memory requirements of an n-gram language model. Quantization of a 3-gram back-off model was performed by applying the *k-means* Lloyd-Max algorithm at each n-gram level. Experiments were performed on several large-vocabulary speech recognition tasks by considering different levels of compression. By encoded probabilities in 4 bits, the increase in word-error-rate was only around 2% relative with respect to a baseline using 32-bit floating point probabilities.

Similar work was carried out in the field of information retrieval, where memory efficiency is instead related to the indexing data structure, which contains information about frequencies of terms in all the individual documents. Franz and McCarley (2002) investigated quantization of term frequencies by applying a binning method. The impact on retrieval performance was analyzed against different quantization levels. Results showed that 2 bits are sufficient to encode term frequencies at the cost of a negligible loss in performance.

In our work, we investigate both data compression methods, namely the Lloyd's algorithm and the binning method, in a SMT framework.

3 Quantization

Quantization provides an effective way of reducing the number of bits needed to store floating point variables. The quantization process consists in partitioning the real space into a finite set of k *quantization levels* and identifying a center c_i for each level, $i = 1, \dots, k$. A function $q(x)$ maps any real-valued point x onto its unique center c_i. Cost of quantization is the approximation error between x and c_i.

If $k = 2^h$, h bits are enough to represent a floating point variable; as a floating point is usually encoded in 32 bits (4 byte), the *compression ratio* is equal to $32/h$[1] . Hence, the compression ratio also gives an upper bound for the relative reduction of memory use, because it assumes an optimal implementation of data structures without any memory waste. Notice that memory consumption for storing the k-entry codebook is negligible ($k * 32$ bits).

As we will apply quantization on probabilistic distribution, we can restrict the range of real values between 0 and 1. Most quantization algorithms require a fixed (although huge) amount of points in order to define the quantization levels and their centers. Probabilistic models used in SMT satisfy this requirement because the set of parameters larger than 0 is always limited.

Quantization algorithms differ in the way partition of data points is computed and centers are identified. In this paper we investigate two different quantization algorithms.

Lloyd's Algorithm

Quantization of a finite set of real-valued data points can be seen as a clustering problem. A large family of clustering algorithms, called *k-means* algorithms (Kanungo et al., 2002), look for optimal *centers* c_i which minimize the mean squared distance from each data point to its nearest center. The map between points and centers is trivially derived.

[1] In the computation of the compression ratio we take into account only the memory needed to store the probabilities of the observations, and not the memory needed to store the observations themselves which depends on the adopted data structures.

As no efficient exact solution to this problem is known, either polynomial-time approximation or heuristic algorithms have been proposed to tackle the problem. In particular, Lloyd's algorithm starts from a feasible set of centers and iteratively moves them until some convergence criterion is satisfied. Finally, the algorithm finds a local optimal solution. In this work we applied the version of the algorithm available in the K-MEANS package[2].

Binning Method

The binning method partitions data points into uniformly populated intervals or *bins*. The center of each bin corresponds to the mean value of all points falling into it. If N_i is the number of points of the i-th bin, and x_i the smallest point in the i-th bin, a partition $[x_i, x_{i+1}]$ results such that N_i is constant for each $i = 0, \dots, k-1$, where $x_k = 1$ by default. The following map is thus defined:

$$q(x) = c_i \text{ if } x_i <= x < x_{i+1}.$$

Our implementation uses the following *greedy* strategy: bins are build by uniformly partition all different points of the data set.

4 Phrase-based Translation System

Given a string $\mathbf{f}$ in the source language, our SMT system (Federico and Bertoldi, 2005; Cettolo et al., 2005), looks for the target string $\mathbf{e}$ maximizing the posterior probability $\Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f})$ over all possible word alignments $\mathbf{a}$. The conditional distribution is computed with the log-linear model:

$$p_{\boldsymbol{\lambda}}(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \quad \propto \quad \exp\left\{\sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{a})\right\},$$

where $h_r(\mathbf{e}, \mathbf{f}, \mathbf{a}), r = 1 \dots R$ are real valued feature functions.

The log-linear model is used to score translation hypotheses $(\mathbf{e}, \mathbf{a})$ built in terms of strings of phrases, which are simple sequences of words. The translation process works as follows. At each step, a target phrase is added to the translation whose corresponding source phrase within $\mathbf{f}$ is identified through three random quantities: the *fertility* which establishes its length; the *permutation* which sets its first position; the *tablet* which tells its word string. Notice that target phrases might have fertility equal to zero, hence they do not translate any source word. Moreover, untranslated words in $\mathbf{f}$ are also modeled through some random variables.

The choice of permutation and tablets can be constrained in order to limit the search space until performing a monotone phrase-based translation. In any case, local word reordering is permitted by phrases.

The above process is performed by a beam-search decoder and is modeled with twelve feature functions (Cettolo et al., 2005) which are either estimated from data, e.g. the target n-gram language models and the phrase-based translation model, or empirically fixed, e.g. the permutation models. While feature functions exploit statistics extracted from monolingual or word-aligned texts from the training data, the scaling factors λ of the log-linear model are empirically estimated on development data.

The two most memory consuming feature functions are the phrase-based Translation Model (TM) and the n-gram Language Model (LM).

Translation Model

The TM contains phrase-pairs statistics computed on a parallel corpus provided with word-alignments in both directions. Phrase-pairs up to length 8 are extracted and singleton observations are pruned off. For each extracted phrase-pair $(\tilde{f}, \tilde{e})$, four translation probabilities are estimated:
– a smoothed frequency of $\tilde{f}$ given $\tilde{e}$
– a smoothed frequency of $\tilde{e}$ given $\tilde{f}$
– an IBM model 1 based probability of $\tilde{e}$ given $\tilde{f}$
– an IBM model 1 based probability of $\tilde{f}$ given $\tilde{e}$

Hence, the number of parameters of the translation models corresponds to 4 times the number of extracted phrase-pairs. From the point of view of quantization, the four types of probabilities are considered separately and a specific codebook is generated for each type.

Language Model

The LM is a 4-gram back-off model estimated with the *modified Kneser-Ney smoothing* method (Chen and Goodman, 1998). Singleton pruning is applied on 3-gram and 4-gram statistics. In terms of num-

[2] www.cs.umd.edu/∼mount/Projects/KMeans.

task	parallel resources		mono resources	LM				TM
	src	trg	words	1-gram	2-gram	3-gram	4-gram	phrase pairs
NIST	82,168	88,159	463,855	1,408	20,475	29,182	46,326	10,410
EPPS	34,460	32,951	3,2951	110	2,252	2,191	2,677	3,877
EPPS-800	23,611	22,520	22,520	90	1,778	1,586	1,834	2,499
EPPS-400	11,816	11,181	11,181	65	1,143	859	897	1,326
EPPS-200	5,954	5,639	5,639	47	738	464	439	712
EPPS-100	2,994	2,845	2,845	35	469	246	213	387

Table 1: Figures (in thousand) regarding the training data of each translation task.

ber of parameters, each n-gram, with $n < 4$, has two probabilities associated with: the probability of the n-gram itself, and the back-off probability of the corresponding $n + 1$-gram extensions. Finally, 4-grams have only one probability associated with.

For the sake of quantization, two separate codebooks are generated for each of the first three levels, and one codebook is generated for the last level. Hence, a total of 7 codebooks are generated. In all discussed quantized LMs, unigram probabilities are always encoded with 8 bits. The reason is that unigram probabilities have indeed the largest variability and do not contribute significantly to the total number of parameters.

5 Experiments

Data and Experimental Framework

We performed experiments on two large vocabulary translation tasks: the translation of European Parliamentary Plenary Sessions (EPPS) (Vilar et al., 2005) from Spanish to English, and the translation of documents from Chinese to English as proposed by the NIST MT Evaluation Workshops[3].

Translation of EPPS is performed on the so-called final text editions, which are prepared by the translation office of the European Parliament. Both the training and testing data were collected by the TC-STAR[4] project and were made freely available to participants in the 2006 TC-STAR Evaluation Campaign. In order to perform experiments under different data sparseness conditions, four subsamples of the training data with different sizes were generated, too.

task	sentences	src words	ref words
EPPS	840	22725	23066
NIST	919	25586	29155

Table 2: Statistics of test data for each task.

Training and test data used for the NIST task are available through the Linguistic Data Consortium[5]. Employed training data meet the requirements set for the Chinese-English large-data track of the 2005 NIST MT Evaluation Workshop. For testing we used instead the NIST 2003 test set.

Table 1 reports statistics about the training data of each task and the models estimated on them. That is, the number of running words of source and target languages, the number of n-grams in the language model and the number phrase-pairs in the translation model. Table 2 reports instead statistics about the test sets, namely, the number of source sentences and running words in the source part and in the gold reference translations.

Translation performance was measured in terms of BLEU score, NIST score, word-error rate (WER), and position independent error rate (PER). Score computation relied on two and four reference translations per sentence, respectively, for the EPPS and NIST tasks. Scores were computed in case-insensitive modality with punctuation. In general, none of the above measures is alone sufficiently informative about translation quality, however, in the community there seems to be a preference toward reporting results with BLEU. Here, to be on the safe side and to better support our findings we will report results with all measures, but will limit discussion on performance to the BLEU score.

In order to just focus on the effect of quantiza-

[3] www.nist.gov/speech/tests/mt/.

[4] www.tc-star.org

[5] www.ldc.upenn.edu

	LM-h						
	32	8	6	5	4	3	2
32	54.78	54.75	54.73	54.65	54.49	54.24	53.82
8	54.78	54.69	54.69	54.79	54.55	54.18	53.65
6	54.57	54.49	54.76	54.57	54.63	54.26	53.60
TM-h 5	54.68	54.68	54.56	54.61	54.60	54.10	53.39
4	54.37	54.36	54.47	54.44	54.23	54.06	53.26
3	54.28	54.03	54.22	53.96	53.75	53.69	53.03
2	53.58	53.51	53.47	53.35	53.39	53.41	52.41

Table 3: BLEU scores in the EPPS task with different quantization levels of the LM and TM.

tion, all reported experiments were performed with a plain configuration of the ITC-irst SMT system. That is, we used a single decoding step, no phrase re-ordering, and task-dependent weights of the log-linear model.

Henceforth, LMs and TM quantized with h bits are denoted with `LM-h` and `TM-h`, respectively. Non quantized models are indicated with `LM-32` and `TM-32`.

Impact of Quantization on LM and TM

A first set of experiments was performed on the EPPS task by applying probability quantization either on the LM or on the TMs. Figures 1 and 2 compare the two proposed quantization algorithms (`LLOYD` and `BINNING`) against different levels of quantization, namely 2, 3, 4, 5, 6, and 8 bits. The scores achieved by the non quantized models (`LM-32` and `TM-32`) are reported as reference.

The following considerations can be drawn from these results. The Binning method works slightly, but not significantly, better than the Lloyd's algorithm, especially with the highest compression ratios.

In general, the LM seems less affected by data compression than the TM. By comparing quantization with the binning method against no quantization, the BLEU score with LM-4 is only 0.42% relative worse (54.78 vs 54.55). Degradation of BLEU score by TM-4 is 0.77% (54.78 vs 54.36). For all the models, encoding with 8 bits does not affect translation quality at all.

In following experiments, binning quantization was applied to both LM and TM. Figure 3 plots all scores against different levels of quantization. As references, the curves corresponding to only LM quantization (`LM-h`) and only TM quantization (`TM-h`) are shown. Independent levels of quantization of the LM and TM were also considered. BLEU scores related to several combinations are reported in Table 3.

LM-h	TM-h	BLEU	NIST	WER	PER
32	32	28.82	8.769	62.41	42.30
8	8	28.87	8.772	62.39	42.19
4	4	28.36	8.742	62.94	42.45
2	2	25.95	8.491	65.87	44.04

Table 4: Translation scores on the NIST task with different quantization levels of the LM and TM.

Results show that the joint impact of LM and TM quantization is almost additive. Degradation with 4 bits quantization is only about 1% relative (from 54.78 to 54.23). Quantization with 2 bits is surprisingly robust: the BLEU score just decreases by 4.33% relative (from 54.78 to 52.41).

Quantization vs. Data Sparseness

Quantization of LM and TM was evaluated with respect to data-sparseness. Quantized and not quantized models were trained on four subset of the EPPS corpus with decreasing size. Statistics about these sub-corpora are reported in Table 1. Quantization was performed with the binning method using 2, 4, and 8 bit encodings. Results in terms of BLEU score are plotted in Figure 4. It is evident that the gap in BLEU score between the quantized and not quantized models is almost constant under different training conditions. This result suggests that performance of quantized models is not affected by data sparseness.

Consistency Across Different Tasks

A subset of quantization settings tested with the EPPS tasks was also evaluated on the NIST task. Results are reported in Table 4.

Quantization with 8 bits does not affect performance, and gives even slightly better scores. Also quantization with 4 bits produces scores very close to those of non quantized models, with a loss in BLEU score of only 1.60% relative. However, pushing quantization to 2 bits significantly deteriorates performance, with a drop in BLEU score of 9.96% relative.

In comparison to the EPPS task, performance degradation due to quantization seems to be twice as large. In conclusion, consistent behavior is observed among different degrees of compression. Absolute loss in performance, though quite different from the EPPS task, remains nevertheless very reasonable.

Performance vs. Compression

From the results of single versus combined compression, we can reasonably assume that performance degradation due to quantization of LM and TM probabilities is additive. Hence, as memory savings on the two models are also independent we can look at the optimal trade-off between performance and compression separately. Experiments on the NIST and EPPS tasks seem to show that encoding of LM and TM probabilities with 4 bits provides the best trade-off, that is a compression ratio of 8 with a relative loss in BLEU score of 1% and 1.6%. It can be seen that score degradation below 4 bits grows generally faster than the corresponding memory savings.

6 Conclusion

In this paper we investigated the application of data compression methods to the probabilities stored by a phrase-based translation model. In particular, probability quantization was applied on the n-gram language model and on the phrase-pair translation model. Experimental results confirm previous findings in speech recognition: language model probabilities can be encoded in just 4 bits at the cost of a very little loss in performance. The same resolution level seems to be a good compromise even for the translation model. Remarkably, the impact of quantization on the language model and translation model seems to be additive with respect to performance. Finally, quantization does not seems to be affected by data sparseness and behaves similarly on different translation tasks.

References

M Cettolo, M. Federico, N. Bertoldi, R. Cattoni, and B. Chen. 2005. A Look Inside the ITC-irst SMT System. In *Proc. of MT Summit X*, pp. 451–457, Pukhet, Thailand.

S. F. Chen and J. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, USA.

M. Federico and N. Bertoldi. 2005. A Word-to-Phrase Statistical Translation Model. *ACM Transaction on Speech Language Processing*, 2(2):1–24.

M. Franz and J. S. McCarley. 2002. How Many Bits are Needed to Store Term Frequencies. In *Proc. of ACM SIGIR*, pp. 377–378, Tampere, Finland.

J. Gao and M. Zhang. 2002. Improving Language Model Size Reduction using Better Pruning Criteria. In *Proc. of ACL*, pp. 176–182, Philadelphia, PA.

T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, , and A. Y. Wu. 2002. An Efficient K-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(7):881–892.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of HLT/NAACL 2003*, pp. 127–133, Edmonton, Canada.

F. J. Och and H. Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

B. Raj and E. W. D. Whittaker. 2003. Lossless Compression of Language Model Structure and Word Identifiers. In *Proc. of ICASSP*, pp. 388–391, Honk Kong.

K. Seymore and R. Rosenfeld. 1996. Scalable Backoff Language Models. In *Proc. of ICSLP*, vol. 1, pp. 232–235, Philadelphia, PA.

D. Vilar, E. Matusov, S. Hasan, R . Zens, , and H. Ney. 2005. Statistical Machine Translation of European Parliamentary Speeches. In *Proc. of MT Summit X*, pp. 259–266, Pukhet, Thailand.

E. W. D. Whittaker and B. Raj. 2001. Quantization-based Language Model Compression. In *Proc. of Eurospeech*, pp. 33–36, Aalborg, Denmark.

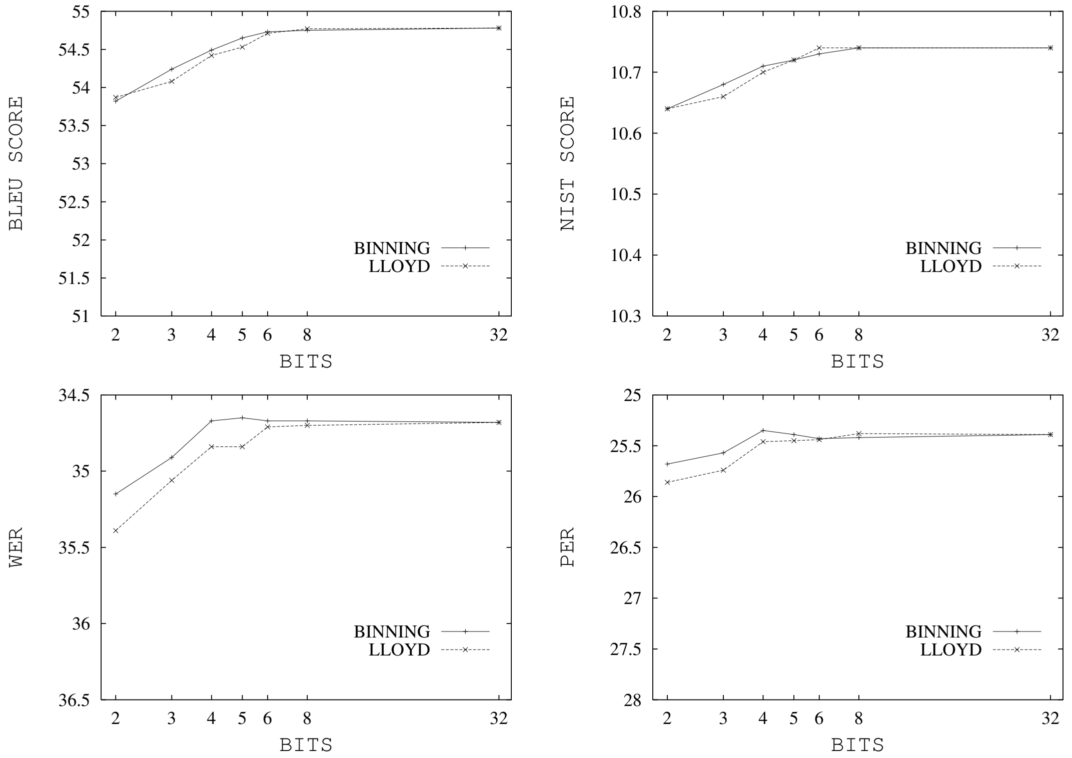

Figure 1: EPPS task: translation scores vs. quantization level of LM. TM is not quantized.

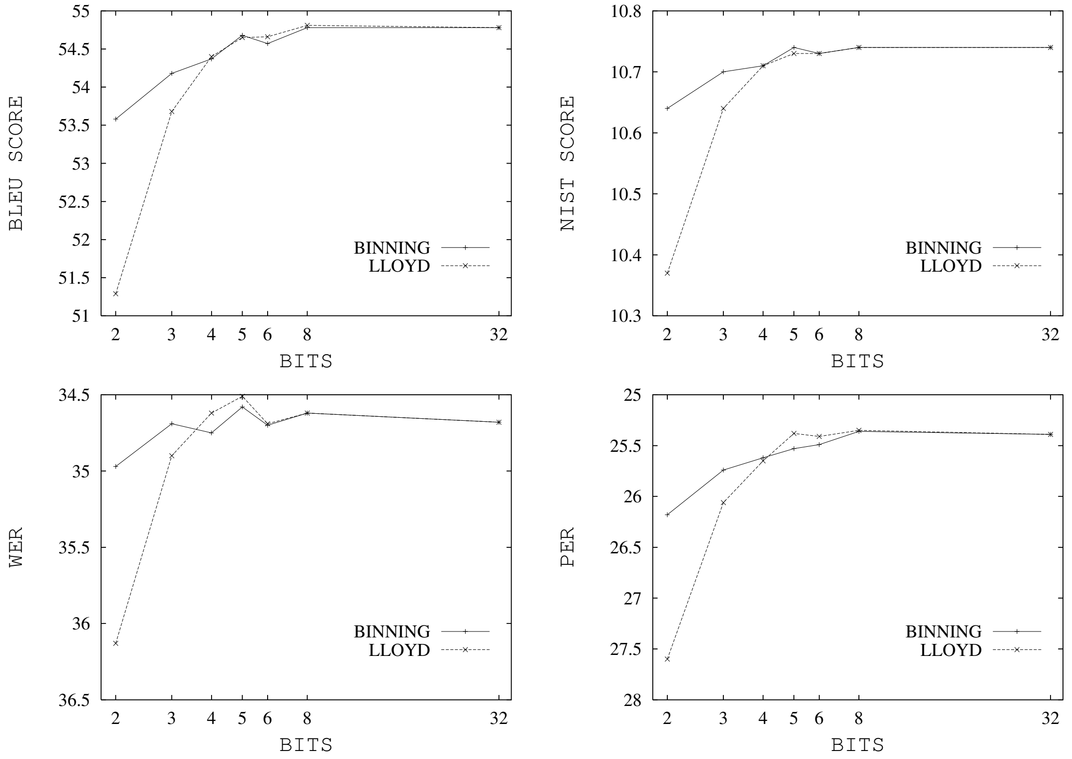

Figure 2: EPPS task: translation scores vs. quantization level of TM. LM is not quantized.

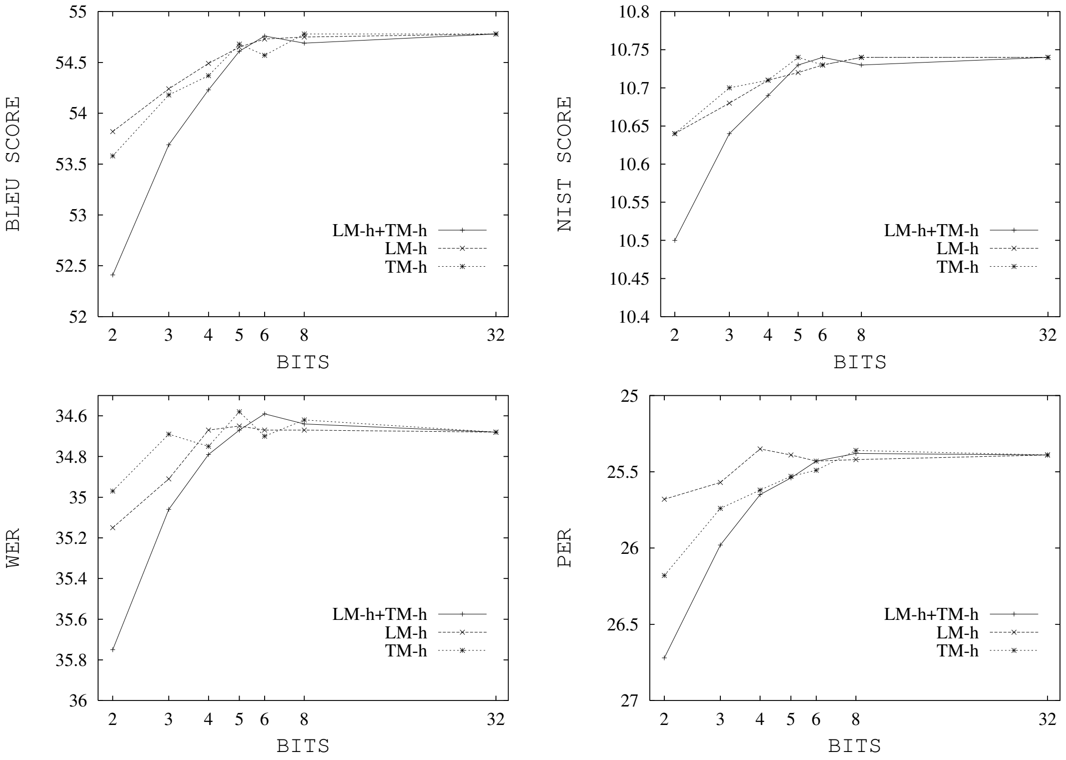

Figure 3: EPPS task: translation scores vs. quantization level of LM and TM. Quantization was performed with the Binning algorithm.

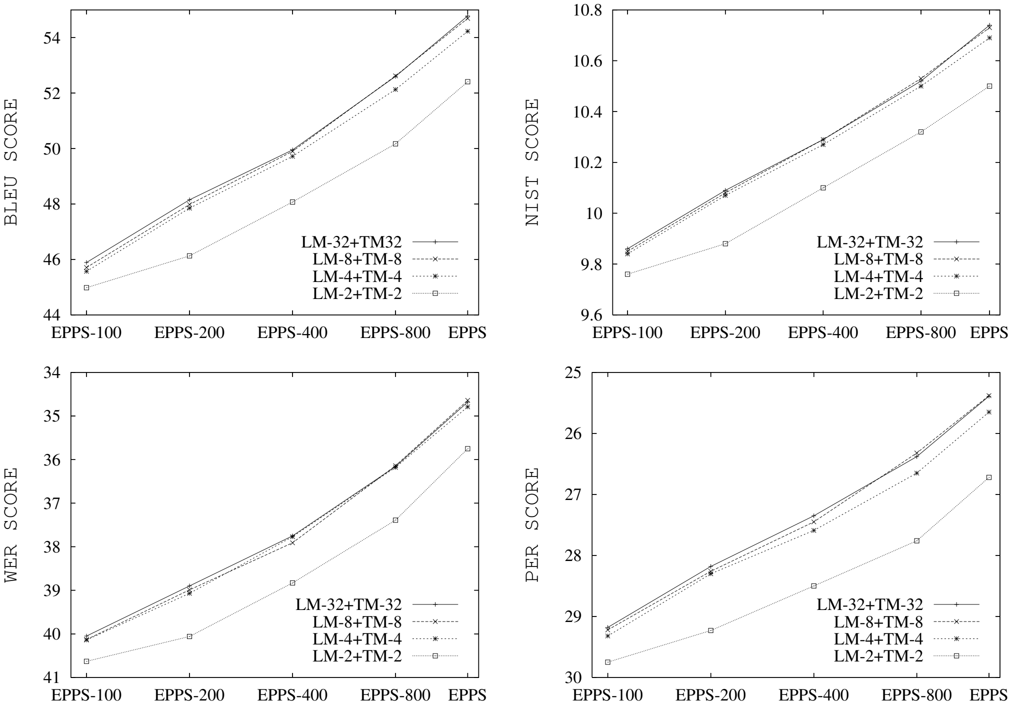

Figure 4: EPPS task: translation scores vs. amount of training data. Different levels of quantization were generated with the Binning algorithm.