The QMUL System Description for IWSLT 2008

Simon Carter, Christof Monz, Sirvan Yahyaei

Computer Science Department
Queen Mary, University of London
London, E1 4NS, UK
{simonc, christof, sirvan}@dcs.qmul.ac.uk

Abstract

The QMUL system to the IWSLT 2008 evaluation campaign is a phrase-based statistical MT system implemented in C++. The decoder employs a multi-stack architecture, and uses a beam to manage the search space. We participated in both BTEC Arabic → English and Chinese → English tracks, as well as the PIVOT task. In our first submission to IWSLT, we are particularly interested in seeing how our SMT system performs with speech input, having so far only worked with and translated newswire data sets.

1. Introduction

The IWSLT 2008 evaluation campaign has allowed us to gain experience working with spoken language translation. Three systems were submitted for evaluation: Arabic → English and Chinese → English BTEC tasks and the Chinese → Spanish PIVOT task.

In Section 2.1 we look at phrase-based SMT, before going on to describe the features used in out system in Section 2.2 and describe our decoder in Section 2.3. We then go on to look at the experimental set up and the data used in the in Section **??**, before presenting the results of our system in Section 3.

2. Translation Framework

The aim of Statistical Machine Translation(SMT) is to take a foreign sentence, **f**, and translate it into an English sentence, **e** using statistical models generated using machine learning techniques. Using a corpus of foreign and target sentences which we know to be translations of one another, SMT becomes a problem of constructing accurate probability distributions, or models, that can be used to translate a collection of foreign sentences, unseen in the training data, into a target language. In this section we start by describing phrase based translation, before examining the models we use, and then describing the QMUL decoder.

2.1. Phrase Based SMT

Intuitively, given a source sentence **f**, the problem of statistical machine translation can be formulated as picking the target sentence **e** with the highest probability according to the model $Pr(\mathbf{e}|\mathbf{f})$. Using Bayes theorem, we can say:

$$Pr(\mathbf{e}|\mathbf{f}) = \frac{Pr(\mathbf{f}|\mathbf{e})Pr(\mathbf{e})}{Pr(\mathbf{f})} \quad (1)$$

Given that we seek to find the target, or English sentence for which the probability arrived at through equation 1 is greatest, and considering that the denominator $Pr(\mathbf{f})$ is a constant independent of **e**, we can reformulate equation 1 to:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})\} \quad (2)$$

This approach is often referred to as the noisy channel approach [1]. $p(\mathbf{f}|\mathbf{e})$ is the translation model, or the likelihood of generating the source sentence given the target sentence, and **e** is the language model, which tells us the likelihood of a given English sentence. Together these form the core of all SMT systems, and thus equation 2 is described as the Fundamental Equation of SMT [1].

Phrase based SMT, as described by [2, 3], extends the noisy channel approach by using a weighted log linear combination of a set of H feature functions, $h_i, i = 1, ..., H$, to score possible translations.

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e},\mathbf{a}} \sum_{i=1}^{H} \lambda_i h_i(\mathbf{f}, \mathbf{e}) \quad (3)$$

$p(\mathbf{f}|\mathbf{e})$ and $p(\mathbf{e})$ become a subset of the H different feature functions, each with their own weights λ_i, which can be trained according to an optimisation criterion based on the translation quality [4], as we have done for IWSLT 08.

2.2. Features

In addition to the phrase translation probabilities that have already been discussed, there are a number of common features that we use during decoding. As these features operate over phrase segmentations, it is first usefull to define what a phrase is. If we represent a given sentence pair as (f_1^J, e_1^I), it's phrase segmentation can be defined in terms of K units:

$$k \rightarrow s_k := (i_k; b_k, j_k), \text{ for } k = 1...K \quad (4)$$

(b_k, j_k) denotes the start and end positions of the source phrase that is aligned to the k^{th} target phrase, and i_k denotes the last position of the k^{th} target phrase. Because the models operate over the phrase segmentations k in s, we say the model features are over (f_1^J, e_1^I, s_1^K). A more detailed description of the phrase segmentation can be found in [5].

Our language model is a standard n-gram based feature function, where the probability of a given word is conditioned on its history:

$$h_{LM}(f_1^J, e_1^I, s_1^K) = \log \prod_{i=1}^{l+1} p(e_i | e_{i-n+1}^{i-1}) \quad (5)$$

We used the publicly available SRILM toolkit [6] to construct tri-gram Kneser-Ney discounted language models.

Re-ordering issues are explicitly dealt with through a simple feature that penalises jumps during decoding with respect to the jump width. Thus long distance re-ordering, which requires a long jump within the source sentence, is penalised more heavily than smaller, more local re-ordering. If ep represents the end position of the last phrase and sp is the start position of the new phrase, then:

$$h_{RM}(f_1^J, e_1^I, s_1^K) = -\sum_{k=1}^{K} |sp_k - ep_{k-1} - 1| \quad (6)$$

Due to the reliance on relative frequencies by phrase translation probabilities, longer phrases which tend to be rare have over estimated probabilities. We therefore use a lexicon model, based on the frequencies of the words within a phrase, to smooth the phrase translation probabilities. As with the phrase based model, the word based lexicon is used in both translation directions $p(\mathbf{f}|\mathbf{e})$ and $p(\mathbf{e}|\mathbf{f})$.

$$h_{LEX}(f_1^J, e_1^I, s_1^K) = \log \prod_{k=1}^{K} \prod_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} p(f_j | e_i) \quad (7)$$

To control the length of the translation, we use a word penalty, described in Equation 8. A negative penalty favours longer translations, and positive penalties are used to produce shortened translations.

$$h_{WP}(f_1^J, e_1^I, s_1^K) = I \quad (8)$$

Similar to the word penalty, we use a phrase penalty to control the number of phrase applications used in the translation of a source sentence:

$$h_{PP}(f_1^J, e_1^I, s_1^K) = K \quad (9)$$

K is the number of phrase segmentation used. As with the word penalty h_{WP}, if longer phrases are trusted over smaller phrases, which is intuitively the case as the longer phrase carries more context, than a positive penalty can be used. If smaller phrase applications are desired, a negative penalty would be employed.

In our system we also use a binary version of a phrase count feature, which favours phrase segmentations that appear in the bitext over a certain threshold. Generally rare phrase pairs have overestimated probabilities. By taking into account the phrase count, we can make up for data that may just be noise, representing mistranslations or erratic word alignments.

$$h_{c_r}(f_1^J, e_1^I, s_1^K) = \sum_{K=1}^{K} [N(f_k, e_k) \leq r] \quad (10)$$

The feature is binary, so that a phrase has a cost of 0 or 1 in log space, dependent on whether the threshold r is met or not. For our experiments an r value of 4 was manually chosen. For a more detailed description of these models please refer to [7].

2.3. Decoding

The QMUL system is a stack based decoder implemented in C++, similar to the publicly available Pharaoh system [8]. Hypotheses are ranked and stored in stacks, where each stack represents the number of source words translated so far. We start with a null hypothesis, and pick a segment of the source sentence to begin translating. We apply multiple different translations that exist for the same source phrase, and then store the hypothesis in a stack. The hypotheses in a stack are ranked according to their score, which takes into account the phrase translation costs, language model cost, and other feature functions in the system mentioned in section 2.2. The retrieved translation is the cheapest hypothesis in the final stack representing hypotheses with no untranslated foreign words.

As the search space is quadratic with respect to input source sentence size, we manage the number of hypotheses we generate through the use of a beam and stack limit. The beam is set with respect to the highest scoring states in each stack. The stack limit reserves a certain amount of places for the highest scoring hypothesis, meaning that each time we start to expand states in a new stack, we always have the same number. These measures can cause search errors, however increasing the stack limit or widening the beam makes relatively minor improvements to translation quality whilst often dramatically degrading translation speed.

One risk free manner to reduce the search space is through hypothesis re-combination. If two states share the following same characteristics, then they can be safely combined into one:

- the last two English words generated
- the foreign words covered so far
- the last foreign segment translated

Corpus	Task	BLEU ASR.1	BLEU CCR
BTEC	AE	25.55	28.97
	CE	20.10	22.19
PIVOT	CES	12.47	3.85

Table 1: Primary run results.

If two hypotheses share the same properties, the best scoring one is kept for further analysis, with the other safely discarded.

When comparing the costs of states for pruning purposes, it is necessary to take into account the work they have done and the work yet to do. It would not be fair to discount a state in comparison to another for having translated a difficult part of the source sentence. We therefore include a future cost, calculated prior to the translation process using a dynamic programming algorithm. For each possible translation span in the source sentence, we take the product of the translation and language model probabilities. We then store these costs in a table for lookup during decoding.

During decoding, if a hypothesis has non-contiguous segments of untranslated source phrases, the future cost is then the product of the future costs of each segment retrieved from the look up table. Notice that while we assume monotonic decoding when pre-computing the future cost matrix, when calculating the actual future cost for a state, we include the best possible distortion model cost involved in translating the uncovered source sentence segments.

3. Experiments

Table 1 displays the QMUL results for IWSLT 08. The poor results reflect the fact that we had little time to prepare our submissions to the IWSLT08 evaluation campaign. In particular, the system for the PIVOT was not optimised, and results for the PIVOT CCR task are influenced by errors in our implementation.

The experiments were conducted using the provided data sets from the Basic Travel Expression Corpus (BTEC). The BTEC corpus contains relatively short tourism related sentences. For the BTEC and PIVOT tasks, one set of training data comprising just 20k aligned sentences and six development sets from previous IWSLT conferences were provided. Corpus statistics can be seen in Table 2. For the Arabic and Chinese BTEC tasks we only used the bitext supplied to us by IWSLT for training purposes. Word alignment was conducted using the open source GIZA++ toolkit [9], and phrase extraction is done according to Philip Koehn's refined alignment and extraction software [10].

We were provided with 6 different development sets. To be able to compare our systems during tuning to previous IWSLT submissions, we optimised our systems on devset5 with 7 references per translation, and evaluated on devset6 with 6 references per translation. Optimisation was carried out using a minimum error rate trainer (MERT).

BTEC AE BITEXT	**Arabic**	**English**
Sentences	19723	
Tokens	14649	6778
BTEC CE BITEXT	**Chinese**	**English**
Sentences	19723	
Tokens	8387	7716
PIVOT CE BITEXT	**Chinese**	**English**
Sentences	19723	
Tokens	8387	7716
PIVOT ES BITEXT + EUROPARL	**English**	**Spanish**
Sentences	122088	
Tokens	31653	54228

Table 2: Corpus statistics.

LM	Perplexity	BLEU ASR.1	BLEU CCR
bitext	155.758	0.2706	0.3751
+europarl	151.631	**0.2978**	**0.4235**
+europarl549	151.631	0.2946	0.4201
+bnc	**59.8367**	0.2955	0.4060
+bnc+europarl	94.7524	0.2943	0.4190

Table 3: Perplexity of English LMs tested on the Arabic → English deveset6 references. Bitext is from the English side of supplied training data.

Prior to translating the CCR and ASR development and test sets, we undertook two pre-processing steps to transform aspects of the data, including tokenization and lowercasing. For the Chinese ⟶ English part of the BTEC and PIVOT tasks, Chinese segmentation was left as provided. System output was post processed, including un-tokenizing and true-casing, to make it ready for evaluation.

Our Language Models used for the BTEC and PIVOT tasks were built using the open source SRILM toolkit [6]. Due to the domain of the BTEC corpus, we examined the use of spoken language transcripts from the British National Corpus (BNC). In extracting the spoken language data, we excluded those transcripts where a speech or presentation was being given, thereby enforcing the conversational aspect of the data. There were in total 549K sentences extracted from the BNC corpus. The Europarl data used for both English and Spanish language models totalled 1.3m sentences.

Table 3 shows the perplexities of the various LMs tested on the English side of BTEC Arabic - English bitext, with relevant BLEU scores. The +bnc LM achieved the lowest perplexity, however it was the +europarl LM that received the highest BLEU scores. If we factor out the differences in the LMs resulting from the size of their training corpora, the LM constructed on BNC data performs better, as can be seen in the result of the +europarl549 LM, which was constructed using 549k of Europarl data instead of the full 1.3m. Experiments are conducted with the +bnc LM.

As Arabic is a morphologically rich language, we decided to examine to what effect stripping morphological information would have in improving translation quality when

System	BLEU
Baseline	0.0948
Nopunc	0.1123
Stemmed	0.2598
Stemmed + Nopunc	**0.2955**

Table 4: BLEU scores for ASR.1 Arabic systems evaluated on devset6.

TEXT	System	OOV	BLEU	BLEU CI
CCR	Baseline	16.5%	0.3903	0.4056
	Stemmed	**11.6%**	**0.4060**	**0.4159**
ASR.1	Baseline	63.5%	0.0948	0.0990
	Stemmed	**21.5%**	**0.2598**	**0.2663**

Table 5: Percentage of OOV tokens of baseline and stemmed Arabic systems with BLEU scores (CI: case insensitive). Optimisation was carried out on devset5, and evaluated on devset6.

using a small corpus. We also examined to what effect removing punctuation from the source side of the bitext aids in translating ASR output. Table 4 displays results for the QMUL systems optimised on devset5 and evaluated on devset6.

Stemming Arabic has the greatest biggest single improvement in scores, causing a jump of 15BP over the baseline. An examination of the percentage of out-of-vocabulary (OOV) tokens of the various bitexts can help understand why. Table 5 reports the number of OOV tokens for the baseline and stemmed systems on CCR and ASR.1 best input respectively. We can see that whilst stemming does not play a major factor for CCR text, with an OOV reduction of 5% leading to a marginal increase in BLEU score, for ASR text it does lead to big improvements on word and phrase recognition. Considering the output of arbitrarily chosen sentences from both systems confirms these findings.

Our PIVOT system consisted of two SMT systems, one Chinese $\longrightarrow$ English and the other English $\longrightarrow$ Spanish, used in a piggy-back fashion, where the output of the first was used as input to the second. For this task we were able to augment the IWSLT supplied training data for the English $\longrightarrow$ Spanish system with an extra 100k of aligned sentences from the Europarl corpus. Word alignment and phrase extraction were done in a similar fashion to the BTEC tasks, and optimisation was also carried out using a minimum error rate trainer (MERT). On examination of the cause of the particularly low PIVOT CCR results, we can confirm an error in the configuration file lead to the English $\longrightarrow$ Spanish part of the PIVOT CCR system running without a Language Model.

BTEC	Size	Tokens	Phrase Table Entries
Baseline	30M	14649	320489
Nopunc	28M	14637	304207
Stemmed	28M	9924	316415
Stemmed + nopunc	27M	9912	301387

Table 6: Arabic phrase-table statistics.

4. References

[1] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[2] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of ACL*, Morristown, NJ, USA, 2002, pp. 295–302.

[3] ——, "The alignment template approach to statistical machine translation," *Comput. Linguist.*, vol. 30, no. 4, pp. 417–449, 2004.

[4] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 160–167.

[5] A. Mauser, D. Vilar, G. Leusch, Y. Zhang, and H. Ney, "The RWTH machine translation system for iwslt 2007," in *International Workshop on Spoken Language Translation*, 2007.

[6] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. Int. Conf. Spoken Language Processing (ICSLP 2002)*, 2002.

[7] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of NAACL*, Morristown, NJ, USA, 2003, pp. 48–54.

[8] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, 2004, pp. 115–124.

[9] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[10] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *AMTA*, 2004.