CMPT 413/825 Final Project by Free Whales

Colin J Brown	Tom Dryer	Erik Ross	Justin Lum
301156669	301124984	301115281	301149185
cjbrown@sfu.ca	tdryer@sfu.ca	eross@sfu.ca	jwlum@sfu.ca

Abstract

In this report we describe the implementation and validation of our Chinese to English machine translation system for our CMPT 413/825 course project.

1 Motivation

Like many other challenges in natural language processing, machine translation (MT) is both highly difficult and highly rewarding with many possible real-world applications. In this paper we outline our approach for translation from Chinese to English. Our goal was to combine our decoder and reranker into one MT system and then to improve that system through better handling of unknown Chinese words.

2 Approach

From a high level, our task is to translate Chinese into English, one sentence at a time. Like many other modern MT systems, our system uses phrases, instead of words, as its atomic unit (Wang, 1998; Koehn et al., 2003). A translation model (TM) is used to find likely translations for each Chinese phrase and a language model (LM) is used to ensure the fluency of the english sentences. Our MT system can be logically split into two major components: a reranker and a decoder. The reranker is used to train system the weights. Given these weights and a sentence in Chinese, the decoder attempts to compute the best possible English translation. For any Chinese word in the sentence that is not found in our TM, we attempt to split the word into known subwords in an optimal way. In the following sections we discuss these components in more detail.

2.1 Decoder

Given a Chinese sentence, a TM and an LM, a decoder constructs one or more likely English sentence candidates. A TM consists of lists of possible English translations for a large set of Chinese phrases along with a set of features for each translation. An LM contains the relative frequency of English phrase n-grams (up to 3-grams here) occuring in a corpus which in our case is a set of translated English sentences.

Our decoder uses a log-linear model to rank candidate English translations (Johnson et al., 1999). Given an English sentence, e, a possible Chinese translation, f and a vector of features $\mathbf{h}(\mathbf{e}, \mathbf{f})$, the best translation, $\mathbf{e}^*$, is computed as,

$$\mathbf{e}^*(\mathbf{f}) = \underset{\mathbf{e}}{\mathbf{argmax}} \boldsymbol{\Theta} \cdot \mathbf{h}(\mathbf{e}, \mathbf{f}) \qquad (1)$$

Here, $\mathbf{h}$ consists of 4 TM features, $log(p(e|f))$, $log(p(f|e))$, $log(lex(e|f))$, $log(lex(f|e))$ and one LM feature, $log(p(e))$. Θ is the feature weight vector that is learned using the reranker (discussed below).

The decoder searches for the best sentence translation, $\mathbf{e}^*$, using a beam search algorithm (Tillmann and Ney, 2003). For a given Chinese sentence, $\mathbf{f}$, a set of stacks is created; one stack for each word to be translated. The ith stack is actually a dictionary of candidate translations of i Chinese words. Each candidate translation of i words is stored as a hypothesis structure containing the phrase features, the last phrase of english words in the translation and a pointer to the hypothesis with the previous phrase.

Chinese words may be selected out of order to allow for word rearrangement. A bitmap of length $|\mathbf{f}|$ is used to keep track of which Chinese words have been translated. Each hypothesis in a stack is indexed by such a bitmap representing the Chinese words that it has translated. Thus, for a given subset of translated Chinese words, we only keep one hypothesis, containing the translation with the highest probability.

The one exception to this stack indexing scheme is the last stack which we require to be at least

of size n so that we can output multiple sentence translation candidates (see below). In this case, instead of indexing on the the (full) bitmap, we index on the last phrase and the second last bitmap.

Beginning with a hypothesis containing only the start symbol, the algorithm builds up successively larger translations and finally selects the most probable translation for which all Chinese words have been translated into English. When searching for a new hypothesis, the algorithm must not only search over untranslated Chinese words, but over Chinese phrases. A new hypothesis is added for each contiguous substring of untranslated Chinese words. Also, for each Chinese phrase, many English phrase translations may be viable. The top k most probable phrases are selected as candidates.

2.2 Reranker

The role of the reranker is to tune the weights of the decoder. Given a ranked list of n-best English translation candidates for each Chinese sentence in the dev corpus, generated by our decoder using feature weights Θ_i, we want to find feature weights Θ_{i+1} such that the generated n-best list matches some 'true' ranking of translation quality. Here, we use the (per sentence) BLEU score (Papineni et al., 2002) between the candidate and a ground-truth translation to define what this 'true' ranking is.

In brief, the BLEU score reports n-gram matching precision between output and reference strings (of n-grams up to size 4) with a penalty for output strings that are too short. Typically, BLEU scores are computed over an entire corpus, but here we compute them between output and reference sentences.

Our reranker is based on the Pairwise Ranked Optimisation (PRO) algorithm (Hopkins and May, 2011). In PRO, the idea is to update the ranking by examining pairs of candidate translation sentences. First, we only want to update weights when the difference in BLEU scores between the two candidates is large enough (i.e. $> \alpha$). Then we check if the difference in BLEU score has the same sign as the difference in decoder score (calculated as a weighted sum of features). If not, the weights are updated using a fraction, η, of the difference between feature vectors. This is a perceptron weight update.

This reweighting is repeated for a number of sample pairs of English translation candidates for each Chinese sentence. The whole process is repeated as 4 epochs in order to find the best weights for the given n-best list.

Note that in practice we only run our reranker over the first 200 sentences. We breifly experimented with using more sentences, randomly sampled from the entire corpus and found it to give better results but we had insufficient time to re-run all tests with these improvments.

2.3 Weight Optimization

The entire training process involves a back-and-forth between the decoder and reranker. Beginning with uniform weights, the decoder is run to produce an n-best list. This n-best list is fed to the reranker to produce new weights. In therory these steps could be repeated until convergence. Here, however, we only perform one further decoder step using wieghts from the decoder due to prohibitively long run-times for computing n-best lists.

2.4 Unknown Chinese Words

A major challenge for any MT system with a finite TM is the problem of unknown words (i.e. source words not in the TM). Similar to the approach of Zhang and Sumita, our solution is to perform segmentation on the unknown word in attempt to find a sequence of known words (Zhang and Sumita, 2008).

During decoding, each input sentence is filtered for Chinese words that do not exist in the TM. Our algorithm searches over all segmentations of the word into sub-words such that each sub-word is in the TM. For each such candidate split, the sub-words are passed to our decoder to be scored. The candidate split with the highest score is used in place of the unknown word. If no such candidate exists, then the unknown word is removed from the sentence.

Note that by using our decoder to score the candidate split, we use the current feature weights in the score function. Thus, as our weights are tuned, the segmentation of unknown words should improve as well.

3 Data

Data for constructing LMs and TMs for our MT system came from the 'Hong Kong Parallel Text' and 'GALE Phase 1 Chinese Newsgroup

Parallel Text' Chinese-English corpuses (Ma, 2004a; Ma and Strassel., 2009). Four different sizes of LMs and TMs were generated by Dr. Anoop Sarkar for the purpose of this work. For all of our experiments we used either the smallest models (`lm/en.tiny.3g.arpa`, `toy/phrase-table/phrase_table.out`) which we refer to as 'tiny' or the largest filtered models (`lm/en.gigaword.3g.filtered.train_dev_test.arpa.gz`, `large/phrase-table/dev-filtered/rules_cnt.final.out`) which we refer to as 'large'.

The (dev) corpus of Chinese and translated English sentences for running our system was sourced from the 'Multiple-Translation Chinese Corpus' parts 1 and 3 (Huang and Doddington, 2002; Ma, 2004b). (Note that for time constraints, we did not get a chance to run our system on the test corpus).

4 Code

For our implementation of this system, we reused submitted code from our decoder in assignment 4 and our reranker in assignment 5. We also used (and heavily modified) the model loading code given for assignment 4 and we used the BLEU score calculation code given for assignment 5.

5 Experimental Setup

In the following experiments our goal is to demonstrate that our MT system outputs good English tranlsations of Chinese sentences which improve with a) more training data and b) segmentation of unknown Chinese words. Thus, we trained and tested our system using both the 'tiny' and 'large' models and with and without segmentation of unknown words. Since four unique ground truth English translations were given, we computed a BLEU score for each translation when testing.

For these tests, we set n, the number of translation candidates and the decoder stack size to 100 and we set k, the maximum number of candidate phrase translations to 25.

6 Results

The table below reports BLEU score mean and SD for all performed experiments using uniform weights (i.e. no reranking) and then after one step of reranking.

Decoder	Tiny	Large (dev)
	Initial (uniform) weights	
Baseline	0.0077±0.0009	0.0468±0.0032
UNK Seg.	0.0082±0.0010	0.0467±0.0028
	Reranked weights	
Baseline	0.0106±0.0010	0.0502±0.0042
UNK Seg.	0.0112±0.0008	0.0403±0.0026

7 Analysis of the Results

As expected, using the large models significantly improved the BLEU scores of the system versus using the tiny models. More translation options to search over in the larger TM and better characterization of fluent English sentences by the larger LM lead to better quality translations overall.

While our sub-segmentation of unknown Chinese words resulted in better BLEU scores using the tiny models, it failed to do so when using the large models. This is likely because named entites (such as names of people) are not handled separately. Because these names are unlikely to occur in the TM, they would either be removed, shortening the output, or segmented by our sub-segmentation alogrithm into incorrect words. Either action would likely lead to a lower BLEU score than simply passing the Chinese word to the output. In the case of the tiny models, likely the segmentation was helpful due to many missing, non-name words from the smaller TM which could be segmented into known words.

We also expected the decoder to improve when using weights tuned by the reranker and it did so in all but one case. It is unclear why the decoder using the unknown word segmentation performed poorer after weight tuning but it could be that reranking caused more names to be included in candidate translations which were then not handled correctly.

8 Future Work

Following this work, we plan to improve our approach for translation of Chinese to English. As mentioned above, our unknown word handling suffered from a lack of handling for names. It is likely that adding such a system on top of unknown word segmentation, as is described by Zhang and Sumita would improve the performance significantly (Zhang and Sumita, 2008).

Due to extremely long run-times of our decoder ($\sim$ 8 to 12 hours for parameters mentioned above), we were unable to perform as many tests nor as

many iterations per test as we would have liked. We implemented a parallel version of our decoder, however this solution was limited the number of cores on a single machine and was not suitable for the 'large' models, due to a multiplicative increase in memory footprint. Moving forward, a cluster based approach to decoding seems to be necessisary to better tune and test our MT system.

We also plan to extend our approach to all other languages and finally solve automated canine-to-human translation.

References

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics.

David Graff Huang, Shudong and George Doddington. 2002. Multiple-translation chinese corpus ldc2002t01. web download.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic unification-based grammars. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 535–541. Association for Computational Linguistics.

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-Volume 1*, pages 48–54. Association for Computational Linguistics.

Xiaoyi Ma and Stephanie Strassel. 2009. Gale phase 1 chinese newsgroup parallel text - part 1 ldc2009t15. web download.

Xiaoyi Ma. 2004a. Hong kong parallel text ldc2004t08. web download.

Xiaoyi Ma. 2004b. Multiple-translation chinese (mtc) part 3 ldc2004t07. web download.

Xiaoyi Ma. 2006. Multiple-translation chinese (mtc) part 4 ldc2006t04. web download.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133.

Ye-Yi Wang. 1998. *Grammar inference and statistical machine translation*. Ph.D. thesis, Carnegie Mellon University.

Ruiqiang Zhang and Eiichiro Sumita. 2008. Chinese unknown word translation by subword resegmentation. In *IJCNLP*, pages 225–232.