Cunei Machine Translation Platform: System Description

Aaron B. Phillips, Ralf D. Brown
Language Technologies Institute
Carnegie-Mellon University
{aphillips,ralf}@cmu.edu

Abstract

In this work we present Cunei, a hybrid, open-source platform for machine translation that models each *example* of a phrase-pair *at run-time* and combines them in dynamic collections. This results in a flexible framework that provides consistent modeling and the use of non-local features.

1 Introduction

In this work we present Cunei [14], a hybrid platform for machine translation that draws upon the depth of research in Example-Based MT (EBMT) and Statistical MT (SMT). In particular, Cunei uses a data-driven approach that extends upon the basic thesis of EBMT–that some examples in the training data are of higher quality or are more relevant than others. Yet, it does so in a statistical manner, embracing much of the modeling pioneered by SMT, allowing for efficient optimization. Instead of using a static model for each phrase-pair, at run-time Cunei models each *example* of a phrase-pair in the corpus with respect to the input and combines them into dynamic collections of examples. Ultimately, this approach provides a more consistent model and a more flexible framework for integration of novel run-time features.

Cunei is open-source software and may freely be downloaded from `http://www.cunei.org/`.

2 Comparing Example-Based and Statistical Machine Translation

Example-Based MT posits translations that closely match previous training examples. Many methods have been employed, but the unifying theme is that the system searches the corpus and extracts examples of translations that have a high degree of similarity with the input. Thus, the simplest representation or model of the translation process *is itself* the training corpus. Examples of translations may be retrieved based on a deeper structure such as the syntax or semantics of a phrase. Alternatively, a shallow approach as described in [2], which only retrieves exact lexical matches, may be employed. In this case, all retrieved examples are lexically equivalent to the input, but still vary according to their surrounding context, alignment probability, genre, location in the corpus, etc. This example-specific knowledge is a crucial component of all EBMT systems, and it is the mechanism by which they select the most appropriate translation.

There is no clear modeling approach favored by EBMT. Scoring is performed on an example-by-example basis. The probability of a particular translation is usually obtained by summing over the scores of all examples that are consistent with the translation. However, this summation makes optimization inefficient. Lacking better alternatives, the example-specific features of an EBMT system are typically tuned by hill-climbing each parameter independently, e.g. as described in [4].

Statistical MT's key contribution to the field has been promoting simple models that can be easily optimized. In both early word-based incarnations and more modern phrase-based systems, the fundamental approach has been to model a limited quantity of information that can adequately represent the translation process in a consistent manner.

SMT originally leveraged the source-channel paradigm using Bayes' rule and a well-structured probabilistic model. However, the log-linear model, introduced to SMT by Berger et al. [1] (and popularized

Mikel L. Forcada, Andy Way (eds.)
Proceedings of the 3rd Workshop on Example-Based Machine Translation, p. 29–36
Dublin, Ireland, November 2009

by Och and Ney [11]), threw away the *a priori* structure in the model and has since become the de-facto modeling approach. Log-linear models not only incorporate all the information of their Bayesian counterparts, but they can be easily extended with more features. Crucially, the system builder does not need to understand how all the features interact. Optimization of the model will automatically determine the relative importance of each feature and account for dependencies. Furthermore, optimization of log-linear models within a translation system is straight-forward. A common technique is Och's [10] minimum error rate training (MERT) which uses linear programming to maximize the objective function. A more recent approach by Smith and Eisner [15] uses an annealing-based method that is more likely to find the global optimum, albeit with more computation.

Due both in part to its origins in the source-channel paradigm and the desire to build simple models, SMT has adopted a static, top-down approach to modeling the translation process. Each combination of a source-phrase and a target-phrase, commonly referred to as a phrase-pair, is represented by one log-linear model that represents the probability that the target-phrase is a translation of the source-phrase. The feature functions for each phrase-pair are calculated over the aggregate of all instances of the the phrase-pair in the training data. Generally, they are the maximum likelihood estimates of some attribute of the phrase-pair in the bilingual corpus where each occurance of the phrase-pair is considered equal. Once the model is constructed the training data can be "thrown away" as it provides no further information.

In its default configuration, the popular open-source software package Moses [7] creates a log-linear model with eight features. Each phrase-pair is assigned five features: two lexical scores estimating the conditional probability in each direction using word-translation probabilities, two maximum-likelihood estimates of the conditional probability in each direction based on the phrasal-alignment links, and a constant phrase penalty to prefer the use of longer phrases. The last three features are applied during decoding: a language model score, a distortion penalty, and a word penalty. Nearly all extant SMT systems have these features at their core because their log-linear combination has been found to perform quite well.

3 Cunei: A Hybrid Approach

Cunei is distinguished from a traditional SMT system in that it delays as much computation as possible until run-time. In particular, translations are not retrieved from a pre-built phrase-table, but rather generated dynamically at run-time. Like in pure EBMT systems, the corpus is available at run-time and features are calculated for each example of a translation separately. But, as in SMT, this information is collected into a single log-linear model that is straight-forward to optimize. Cunei achieves this by dynamically modeling collections of translation examples, as described in §3.4, instead of using static phrase-pairs. This approach has three key advantages:

1. Run-time feature extraction makes it easy to model non-local features dependent on the particular input or surrounding translations. For example, Cunei applies a feature to each translation indicating the number of words to the left and right that are not covered by the translation being considered, but also matched the context from the input. Similarly, document-level context features can also be applied, as in [3], by measuring the cosine or Jensen-Shannon distance [9] between the input document and each document in the corpus where a translation occurs.

2. Generating the translations at run-time produces a more consistent translation model. A traditional SMT system performs alignment and phrase-extraction once when the phrase-table is constructed. Typically this is done through a series of heuristics that determine whether two phrases may legitimately form a phrase-pair. Moses provides multiple heuristics for this very task with the default

method known as 'grow-diag-final'. However, phrase-alignment in reality is not so neatly deterministic. For each source phrase in the parallel corpus, there is some probability (perhaps very small) that it translates to any given target phrase with which it has ever co-occurred. Cunei models the phrase extraction at run-time as part of the translation process with a series of alignment features that determine the probability that a given source phrase and target phrase are aligned. These alignment features are part of the final translation model and during each iteration of optimization the weights for these features are modified. The new weights change the probability distribution over the alignments and have the potential to extract different translation. Thus, the extraction and scoring of translations are not two processes, but form one consistent model (described in §3.2).

3. This approach can efficiently search a larger space of possible translations. Phrase-tables quickly grow quite large–consuming gigabytes of disk space–but only a fraction of the phrases are ultimately used while translating a given test set. This problem is exacerbated even more when one allows for translations with gaps, which we do. By generating translations on-the-fly, Cunei only allocates its computation to modeling translations that are required for the input. First, Cunei looks up translations for phrases that exactly, word-for-word match some span of the input. If the given phrase occurs frequently and Cunei believes it can model the translation well, then we can move on. In situations where this is not sufficient, the range of possible translations is no longer limited *a priori*. Cunei can extend the search space and retrieve similar phrases or generalized templates in order to better model the translation process.

The remainder of this section will describe in detail how translations are constructed at run-time.

3.1 Translation Selection

When given a new input to translate, Cunei searches the source-side of the corpus for phrases that match any sub-section of the input.

In order to accomplish this task, during training Cunei constructs a suffix-array index for each type of sequence present in the parallel corpus. Minimally, the corpus will contain a lexical sequence representing the unaltered surface-strings (or less precisely, 'words'). Lemmas, part-of-speech, or statistical cluster labels may be used to generate alternative types of sequences. The suffix array for each type of sequence is queried with the input to locate matches within the parallel corpus. A match may contain as few as one of the tokens from the input or exactly match the entire input. The collection of corpus matches is stored as a lattice with each element indexed by the span of the input it covers.

The corpus matches are not required to be exact representations of the input. For example, a source phrase retrieved by matching only a part-of-speech sequence may be structurally similar to the input, but it is likely to be semantically unrelated. Matches such as these do not 'as-is' provide valid translations, but they do still contain useful information about the translation process. For each token in a match that does not lexically match the input, a gap is formed. The gaps are projected to the target during phrase-alignment in order to form translation templates. Meta-information about the gap, such as the part-of-speech tag, is preserved in order to aid selection of a valid replacement. These translation templates allow for the formation of novel phrases, but also add risk. As such, if exact phrasal translations are present, they are preferred.

For efficiency, not all matches present in the corpus are retrieved. Each match is scored individually by several feature functions to determine its relevance, e.g. how similar the match is to the input. Matches that have gaps are down-weighted. Matches that have the same context as the input (either sentential or document) are preferred. Typically, around a thousand matches are retrieved from the corpus for each span of the input, but only a few hundred are selected as the most promising and retained for alignment.

3.2 Translation Alignment

After a match is found on the source-side of the corpus, Cunei must determine the target phrase to which it aligns. The alignment is treated as a hidden variable and not specified during training. Instead, Cunei uses statistical methods to induce a phrase-alignment. Ideally, the full alignment process could be carried out dynamically at run-time. Unfortunately, even a simple word-alignment such as IBM Model-1 is too expensive. Instead, we run a word-aligner offline (output from GIZA++ [12] and Berkeley [8] are both supported) and use the word alignments as features for an on-line phrase-alignment.

Each source phrase has some probability of aligning to every possible target phrase within a given sentence. This probability is modeled in Cunei by a series of feature functions over the word-alignments. When a source phrase is aligned to a target phrase, it implies that the remainder of the source sentence that is not specified by the source phrase is aligned to the remainder of the target sentence not specified by the target phrase. Separate features model the probability that the word-alignments for tokens within the phrase are concentrated within the phrase boundaries and that the word-alignments for tokens outside the phrase are concentrated outside the phrase boundaries. Additional features model whether to incorporate unknown words, lexicon-based translation probabilities, and the length ratio between the source and target phrase. This approach is modeled after the work of [16] and [5].

For each instance of a match in the corpus, Cunei uses the feature functions to extract a scored *n*-best list of phrase-alignments. Each possible alignment forms an instance of translation between the source phrase and target phrase.

3.3 Translation Scoring

Of particular importance to Cunei's approach is that each *instance* of a translation from the corpus is scored individually. Most traditional SMT systems model a phrase-pair based on features that are maximum likelihood estimates over all instances of the phrase-pair in the corpus and treat each instance equally. Cunei, on the other hand, models each instance separately with its own log-linear model. This allows for features to be dependent on that particular instance of the translation in the parallel corpus. It explicitly models that some instances of translations are better than others. In the next step, all of the instances in the corpus that result in the same translation phrase-pair will be combined together with a single log-linear model for use during decoding.

Many of the per-instance translation features have already been discussed, as they are usually calculated at the earliest stage at which the information is available. During matching, source-side similarity and contextual features are generated. These features are dependent on both the input and the particular sentence or document in the corpus in which the corpus match was found. During phrase-alignment, additional features are calculated to measure the likelihood of the source phrase aligning to the target phrase *in this particular sentence*. Cunei also applies the traditional SMT features based on a translation's overall frequency in the corpus and a constant phrase penalty, none of which change from instance to instance. The scoring framework is intentionally flexible such that features are not hard-coded and any number of features can be added to the translation model at run-time.

3.4 Translation Combination

Thus far we have shown how each instance of a translation acquires several instance-specific features. However, we also need to take into account and explicitly model the fact that some translations are only represented by a few instances in the corpus and other translations occur frequently. This is achieved by modeling collections of translation instances. The final log-linear model for a translation consists of a count feature indicating how many instances are present in the collection along with a series of constraints that indicate the minimum value for each instance-specific feature. A search is performed

over each instance-specific feature such as the alignment quality, genre, or context to determine the set of constraint values that maximizes the score. Weights determined during optimization indicate the relative importance of each constraint and the match count. Lower constraints will allow for more matches, while higher constraints will likely lead to higher quality phrase-pairs but with a smaller number of occurrences.

3.4.1 Formalism

By formally defining how collections of translations are modeled, we see how Cunei's approach encapsulates everything represented within a typical SMT model while remaining consistent with the main idea behind EBMT.

Given translations t, features f, and weights w, a typical SMT model scores translations with a log-linear model, as shown (converted to real space) in Equation 1.

$$score(t_i) = \prod_k f_{i,k}^{w_k} \tag{1}$$

An EBMT system identifies features for each example e in the corpus. As shown in Equation 2, the total score is calculated by summing over all examples that are consistent with the translation t.

$$score(t_i) = \sum_{\forall e_m \in t_i} \prod_k f_{m,k}^{w_k} \tag{2}$$

Cunei's approach, as shown in Equation 3, maximizes over a particular collection of examples c that are consistent with the translation t. This simultaneously avoids the summation which is difficult to optimize, explicitly models frequency as a first-class feature, and allows for example-dependent features continuing to influence the model. Furthermore, the SMT model in Equation 1 is a special case of this formalism when the features for all examples consistent with a translation are constant.

$$score(t_i) = \max_{\forall c_j \subseteq t_i} |c_j|^{w_{matches}} \prod_k \min_{\forall e_m \in c_j} f_{m,k}^{w_k} \tag{3}$$

3.4.2 Global Features

In the prior description of Moses we explained that each phrase-pair is modeled with two conditional phrase probabilities. Since conditional probabilities are useful, we add two more features to Cunei's simple count of examples: the total number of occurrences of the source phrase, and the total number of occurrences of the target phrase. The weighting scheme is then different, but combined log-linearly, these three features have the same expressive power as Moses' two conditional phrase probabilities. Lastly, Moses' phrase penalty and lexical probabilities are incorporated directly as features on each example that 'coincidentally' are constant for all examples that compose a phrase-pair.

3.5 Optimization

Because–at the end of the day–Cunei uses a log-linear model, we can optimize it using the same approaches developed for SMT. In MERT, after each iteration, a new n-best list is generated with the optimized weights. Due to pruning and the beam search within the decoder, the new weights may yield different translations in the n-best list. In our approach, new weights can also change what translations are found in the corpus and how they are modeled. As a result, the search space is larger, but because we select the constraints that maximize the score, the model is still consistent and possible to optimize with

MERT. In practice, MERT is not an ideal choice when the feature-set becomes large. As such, Cunei's optimization code closely follows [15], which was also developed for SMT and also fits a log-linear model to an objective function. The distinction is that this approach minimizes the expected loss over the entire n-best list, making it more appropriate for larger feature spaces.

4 Evaluation

Cunei's *raison d'être* is to provide a machine translation platform in which researchers can exploit features that dynamically extract information from the input and the corpus at run-time. Currently, Cunei is not yet a finished product and only takes advantage of calculating features dynamically for the the alignment and context of each example. Nonetheless, for Cunei to be of use to researchers, it is important to demonstrate that even in its current state, translation quality is comparable to state-of-the-art MT systems such as Moses.

	Finnish-to-English		**French-to-English**		**German-to-English**	
Types	499,770	84,257	106,862	87,083	273,960	86,671
Tokens	21,492,772	29,744,581	34,979,287	32,001,553	29,730,317	31,156,576
Sentences	1,121,312		1,207,184		1,165,545	

Table 1: Corpora Statistics

To compare Cunei with Moses on several languages, we chose the freely-available Europarl corpus [6], and evaluated the performance on translating from Finnish, French, and German into English. We followed the normal practice of reserving the proceedings from the fourth quarter of 2000 for evaluation and trained each system on the remainder of the parallel corpora. Statistics for each of these three parallel corpora are shown in Table 1. Cunei applied light pre-processing, filtering, and tokenization suitable for Western languages to each corpus; the corpora were subsequently word-aligned by GIZA++ in both directions. As all of the systems translated into English, we built one language model consisting of 243 million words from the Europarl and a portion of the English newswire released by the 2009 Workshop on Statistical Machine Translation.[1] An identical tokenized, word-aligned parallel corpus and language model were provided to Cunei and Moses for each language pair, and the translation systems were trained using their default configurations.

The results of the evaluation are shown in Table 2. Each system was evaluated using BLEU [13], since both Moses and Cunei have built-in support to optimize their parameters toward BLEU using a development set. The development set and a blind test were extracted from the evaluation portion of the Europarl collection. Documents were selected using the same set of dates across all corpora, resulting in the English references consisting of roughly the same text. The development set contained close to 3,000 sentences and the evaluation set was nearly 4,000 sentences, with no overlap.

	Finnish-to-English		**French-to-English**		**German-to-English**	
	Dev	Test	Dev	Test	Dev	Test
Moses	0.2445	0.2361	0.3207	0.3219	0.2746	0.2546
Cunei	**0.2456**	0.2369	**0.3215**	0.3225	**0.2813**	0.2634

Table 2: Evaluation on Europarl using BLEU

[1] `http://www.statmt.org/wmt09/`

Finnish -to- English	*Moses*	mr president, indeed himaren orgy of violence and electoral fraud in local elections in the province, who were living in the region kreikkalaisvähemmistöön.
	Cunei	mr president, indeed in the province of violence and electoral fraud in local elections, which were against the greek minority in the area.
	Reference	madam president, it is quite right that the municipal elections in himara were marked by violence and fraud at the expense of the greek minority living there.
French -to- English	*Moses*	for some reason, i know that my name is not on the attendance register.
	Cunei	for some reason i do not know, my name is not on the attendance register.
	Reference	for some strange reason, my name is missing from the register of attendance.
German -to- English	*Moses*	i would like to criticise the lack of initiatives on the new challenges in the safety of employee participation and in industrial relations.
	Cunei	i share the criticism of initiatives to the new challenges in the field of health and safety, and worker participation in the labour relations.
	Reference	i would like to raise a criticism in connection with the lack of initiatives produced in response to the new challenges we face in health and safety at work, employee participation and labour relations.

Table 3: Comparison of Translation Output

Cunei tops Moses ever so slightly in French-to-English, which is the language pair with the fewest divergences and highest overall scores. Interestingly, performance is also nearly identical on Finnish-to-English even though word order in Finnish is much freer and can require substantial reordering during decoding. This result is even more surprising given that reordering has not (yet) been a significant focus in Cunei and our reordering model is only informed by how frequently and how far phrases are moved during decoding. On the other hand, Moses is using a lexicalized distortion model that is much more sophisticated. Only in German-to-English do we see a distinction between Moses and Cunei. We suspect that the key linguistic divergence affecting translation is German compounding. This results in a greater prevalence of one-to-many alignments which makes phrase alignment and extraction more difficult. Cunei shines due to its run-time phrase-extraction which was able to adapt during optimization and learn appropriate weights for aligning and extracting German-English phrase-pairs from less-than-perfect word-alignments.

Table 3 provides one example of translation output for each language-pair (space restrictions prohibited more examples). The most noticeable difference in the output is that Cunei is more likely to find translations of 'unknown' words and tends to exhibit better word selection. However, in comparison to Moses, this at times comes at the cost of less faithful word ordering.

5 Conclusion

In recent years SMT has dominated the field of machine translation research. Even though SMT in many respects grew from the data-driven focus of EBMT, the concept of modeling each example individually has been lost. In this work we attempt to bridge the gap between EBMT and SMT with the introduction of the Cunei Machine Translation Platform. Cunei is open-source and competitive with state-of-the-art SMT systems like Moses. Furthermore, this platform provides researchers with the ability to easily integrate EBMT-style, example-specific features. We achieve this goal by pushing most of the feature extraction and calculation to run-time which ultimately makes the translation model more consistent and allows our system to adapt and explore a larger space during optimization. Work is ongoing to incorporate other sources of information in Cunei. Ultimately, Cunei should be able to use any available information, be it lexical, syntactic, semantic, grammatical, pragmatic, contextual, etc., to make a case-

by-case selection of the best possible examples in its corpus–fulfilling the goal of true 'data-driven' machine translation.

References

[1] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] Ralf D. Brown. Example-based machine translation in the pangloss system. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 169–174, Copenhagen, Denmark, August 1996.

[3] Ralf D. Brown. Exploiting document-level context for data-driven machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 46–55, Waikiki, USA, October 2008.

[4] Ralf D. Brown, Rebecca Hutchinson, Paul N. Bennett, Jaime G. Carbonell, and Peter Jansen. Reducing boundary friction using translation-fragment overlap. In *Machine Translation Summit IX Proceedings*, pages 24–31, New Orleans, USA, September 2003.

[5] Jae Dong Kim, Ralf Brown, Peter Jansen, and Jaime Carbonell. Symmetric probabilistic alignment for example-based translation. In *Proceedings of the Tenth Annual Conference of the European Assocation for Machine Translation*, pages 153–159, Budapest, Hungary, May 2005.

[6] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X Proceedings*, pages 79–86, Phuket, Thailand, September 2005.

[7] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June 2007.

[8] Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 104–111, New York City, USA, June 2006.

[9] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, January 1991.

[10] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003.

[11] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Philadelphia, USA, July 2002.

[12] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA, July 2002.

[14] Aaron B. Phillips. Sub-phrasal matching and structural templates in example-based mt. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 163–170, Skövde, Sweden, September 2007.

[15] David A. Smith and Jason Eisner. Minimum risk annealing for training log-linear models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 787–794, Sydney, Australia, July 2006.

[16] Stephan Vogel. Pesa: Phrase pair extraction as sentence splitting. In *Machine Translation Summit X Proceedings*, pages 251–258, Phuket, Thailand, September 2005.