The Cunei Machine Translation Platform for WMT '10

Aaron B. Phillips
Carnegie Mellon
Pittsburgh, USA.
aphillips@cmu.edu

Abstract

This paper describes the Cunei Machine Translation Platform and how it was used in the WMT '10 German to English and Czech to English translation tasks.

1 The Cunei Machine Translation Platform

The Cunei Machine Translation Platform (Phillips and Brown, 2009) is open-source software and freely available at `http://www.cunei.org/`. Like Moses (Koehn et al., 2007) and Joshua (Li et al., 2009), Cunei provides a statistical decoder that combines partial translations (either phase pairs or grammar rules) in order to compose a coherent sentence in the target language. What makes Cunei unique is that it models the translation task with a non-parametric model that assesses the relevance of each translation instance.

The process begins by encoding in a lattice all possible contiguous phrases from the input.[1] For each source phrase in the lattice, Cunei locates instances of it in the corpus and then identifies the aligned target phrase(s). This much is standard to most data-driven MT systems. The typical step at this stage is to model a phrase pair by computing relative frequencies over the collection of translation instances. This model for the phrase pair will never change and knowledge of the translation instances can subsequently be discarded. In contrast to using a phrase pair as the basic unit of modeling, Cunei models each translation instance. A distance function, represented by a log-linear model, scores the relevance of each translation instance. Our model then sums the scores of translation instances that predict the same target hypothesis.

The advantage of this approach is that it provides a flexible framework for novel sources of information. The non-parametric model still uses information gleaned over all translation instances, but it permits us to define a distance function that operates over one translation instance at a time. This enables us to score a wide-variety of information represented by the translation instance with respect to the input and the target hypothesis under consideration. For example, we could compute how similar one translation instance's parse tree or morpho-syntactic information is to the input. Furthermore, this information will vary throughout the corpus with some translation instances exhibiting higher similarity to the input. Our approach captures that these instances are more relevant and they will have a larger effect on the model. For the WMT '10 task, we exploited instance-specific context and alignment features which will be discussed in more detail below.

1.1 Formalism

Cunei's model is a hybrid between the approaches of Statistical MT and Example-Based MT. A typical SMT model will score a phrase pair with source s, target t, log features ϕ, and weights λ using a log-linear model, as shown in Equation 1 of Figure 1. There is no prototypical model for EBMT, but Equation 2 demonstrates a reasonable framework where evidence for the phrase pair is accumulated over all instances of translation. Each instance of translation from the corpus has a source s' and target t'. In the most limited case $s = s'$ and $t = t'$, but typically an EBMT system will have some notion of similarity and use instances of translation that do not exactly match the input.

Cunei's model is defined in such a way that we maintain the distance function $\phi(s, s', t', t)$ from the EBMT model, but compute it in a much more efficient manner. In particular, we remove the real-space summation within a logarithm that makes it impractical to tune model weights. However, our

[1] Cunei offers limited support for non-contiguous phrases, similar in concept to grammar rules, but this setting was disabled in our experiments.

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 149–154,
Uppsala, Sweden, 15-16 July 2010.

$$score(s,t) = \sum_k \lambda_k \phi_k(s,t) \tag{1}$$

$$score(s,t) = \ln \sum_{s',t'} e^{\sum_k \lambda_k \phi_k(s,s',t',t)} \tag{2}$$

$$score(s,t) = \delta + \sum_k \lambda_k \left(\frac{\sum_{(s',t') \in C} \phi_k(s,s',t',t) e^{\sum_i \lambda_i \phi_i(s,s',t',t)}}{\sum_{(s',t') \in C} e^{\sum_i \lambda_i \phi_i(s,s',t',t)}} \right) \tag{3}$$

Figure 1: Translation model scores according to SMT (1), EBMT (2), and Cunei (2)

model preserves the first-order derivative of Equation 2, which is useful during optimization to locally approximate the hypothesis space. While the inner term initially appears complex, it is simply the expectation of each feature under the distribution of translation instances and can be efficiently computed with an online update. Last, the introduction of δ, a slack variable, is necessary to additionally ensure that the score of this model is equal to Equation 2. Specifying the model in this manner ties together the two different modeling approaches pursued by SMT and EBMT; the SMT model of Equation 1 is merely a special case of our model when the features for all instances of a translation are constant such that $\phi_k(s,s',t',t) = \phi_k(s,t)\ \forall s',t'$.

Indeed, this distinction illuminates the primary advantage of our model. Each feature is calculated particular to one translation instance in the corpus and each translation instance is scored individually. The model is then responsible for aggregating knowledge across multiple instances of translation. Unlike the SMT model, our aggregate model does not maintain feature independence. Each instance of translation represents a *joint* set of features. The higher the score of a translation instance, the more *all* its features inform the aggregate model. Thus, our model is biased toward feature values that represent relevant translation instances.

1.2 Context

Not all translations found in a corpus are equally useful. Often, when dealing with data of varying quality, training a SMT system on all of the data *degrades performance*. A common workaround is to perform some sort of sub-sampling that selects a small quantity of novel phrase pairs from the large out-of-domain corpus such that they do not overwhelm the number of phrase pairs extracted from the smaller in-domain corpus.

Instead of building our model from a heuristic sub-sample, we utilize Cunei's modeling approach to explicitly identify the relevance of each translation instance. We add features to the model that identify when a translation instance occurs within the same context as the input. This permits us to train on *all* available data by dynamically weighting each instance of a translation.

First, we capture the broader context or genre of a translation instance by comparing the document in the corpus from which it was extracted to the input document. These documents are modeled as a bag of words, and we use common document-level distance metrics from the field of information retrieval. Specifically, we implement as features document-level precision, recall, cosine distance and Jensen-Shannon distance (Lin, 1991).

In order to capture local, intra-sentential context, we compare the words immediately to the left and right of each translation instance with the input. We add one feature that counts the total number of adjacent words that match the input and a second feature that penalizes translation instances whose adjacent context only (or mostly) occurs in one direction. As a variation on the same concept, we also add four binary features that indicate when a *unigram* or *bigram* match is present on the *left* or *right* hand side.

The corpus in which an instance is located can also substantially alter the style of a translation. For example, both the German to English and the Czech to English corpora consisted of in-domain News Commenary and out-of-domain Europarl text. When creating the index, Cunei stores the name of the corpus that is associated with each sentence. From this information we create a set of binary features for each instance of translation that indicate from which corpus the instance originated. The weights for these origin features can be

conceived as mixture weights specifying the relevance of each corpus.

1.3 Alignment

After a match is found on the source-side of the corpus, Cunei must determine the target phrase to which it aligns. The phrase alignment is treated as a hidden variable and not specified during training. Ideally, the full alignment process would 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 and our on-line phrase alignment computes features over the the word alignments. The phrase alignment features are then components of the model for each translation instance. While the calculations are not exactly the same, conceptually this work is modeled after (Vogel, 2005).

For each source-side match in the corpus, an alignment matrix is loaded for the complete sentence in which the match resides. This alignment matrix contains scores for all word correspondences in the sentence pair and can be created using GIZA++ (Och and Ney, 2003) or the Berkeley aligner (Liang et al., 2006). Intuitively, when a source phrase is aligned to a target phrase, this 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 compute 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. In addition, words with no alignment links or weak alignments links demonstrate uncertainty in modeling. To capture this effect, we incorporate two more features that count the number of uncertain alignments present in the source phrase and the target phrase.

The features described above assess the phrase alignment likelihood for a particular translation instance. Because they operate over all the word alignments present in a sentence, the alignment scores are contextual and usually vary from instance to instance. As the model weights change, so too will the phrase alignment scores. Each source phrase is modeled as having some probability of aligning to every possible target phrase within a given sentence. However, it is not practical to compute all possible phrase alignments, so we extract translation instances using only a few high-scoring phrase alignments for each occurrence of a source phrase in the corpus.[2] As discussed previously, these extracted translation instances form the basic modeling unit in Cunei.

1.4 Optimization

Cunei's built-in optimization code closely follows the approach of (Smith and Eisner, 2006), which minimizes the expectation of the loss function over the distribution of translations present in the n-best list. Following (Smith and Eisner, 2006), we implemented $\log(\text{BLEU})$ as the loss function such that the objective function can be decomposed as the expected value of BLEU's brevity penalty and the expected value of BLEU's precision score. The optimization process slowly anneals the distribution of the n-best list in order to avoid local minima. This begins with a near uniform distribution of translations and eventually reaches a distribution where, for each sentence, nearly all of the probability mass resides on the top translation (and corresponds closely with the actual 1-best BLEU score). In addition, Cunei supports the ability to decode sentences *toward* a particular set of references. This is used to prime the optimization process in the first iteration with high-scoring, obtainable translations.

2 The WMT '10 Translation Task

For the WMT '10 Translation Task we built two systems. The first translated from German to English and was trained with the provided News Commentary and Europarl (Koehn, 2005) corpora. The second system translated from Czech to English and used the CzEng 0.9 corpus (Bojar and Žabokrtský, 2009), which is a collection of many different texts and includes the Europarl. To validate our results, we also trained a Moses system with the same corpus, alignments, and language model.

2.1 Corpus Preparation

A large number of hand-crafted regular expressions were used to remove noise (control characters, null bytes, etc.), normalize (hard spaces vs. soft spaces, different forms of quotations,

[2] This is controlled by a score ratio that typically selects 2-6 translation instances per occurrence of a source phrase.

render XML codes as characters, etc.), and tokenize (abbreviations, numbers, punctuation, etc.). However, these rules are fairly generic and applicable to most Western languages. In particular, we did not perform any morphologically-sensitive segmentation. From the clean text we calculated the expected word and character ratios between the source language and the target language. Then we proceeded to remove sentence pairs according to the following heuristics:

- A sentence exceeded 125 words
- A sentence exceeded 1,000 characters
- The square of the difference between the actual and expected words divided by the square of the standard deviation exceeded 5
- The square of the difference between the actual and expected characters divided by the square of the standard deviation exceeded 5

All of these processing routines are included as part of the Cunei distribution and are configurable options. An overview of the resulting corpora is shown in Table 1.

Finally, we used the GIZA++ toolkit (Och and Ney, 2003) to induce word alignments in both directions for each language pair. The resulting corpus and word alignments were provided to Moses and Cunei for training. Each system used their respective phrase extraction and model estimation routines.

2.2 Language Model

We intentionally selected two language pairs that translated into English so that we could share one language model between them. We used the large monolingual English News text made available through the workshop and augmented this with the Xinhua and AFP sections of the English Gigaword corpus (Parker and others, 2009). In all, approximately one billion words of English text were fed to the SRILM toolkit (Stolcke, 2002) to construct a single English 5-gram language model with Kneser-Ney smoothing.

2.3 Experiments

The newswire evaluation sets from the prior two years were selected as development data. 636 sentences were sampled from WMT '09 for tuning and all 2,051 sentences from WMT '08 were reserved for testing. Finally, a blind evaluation was also performed with the new WMT '10 test set. All systems were tuned toward BLEU (Papineni et al., 2002) and all evaluation metrics were run on lowercased, tokenized text.

The results in Table 2 and Table 3 show the performance of Cunei[3] against the Moses system we also built with the same data. The first Cunei system we built included all the alignment features discussed in §1.3. These per-instance alignment features are essential to Cunei's run-time phrase extraction and cannot be disabled. The second, and complete, system added to this all the context features described in §1.2. Cunei, in general, performs significantly better than Moses in German and is competitive with Moses in Czech. However, we hoped to see a larger gain from the addition of the context features.

In order to better understand our results and see if there was greater potential for the context features, we selectively added a few of the features at a time to the German system. These experiments are reported in Table 4. What is interesting here is that most subsets of context features did better than the whole and none degraded the baseline (at least according to BLEU) on the test sets. We did not expect a fully additive gain from the combination, as many of the context features do represent different ways of capturing the same phenomena. However, we were still surprised to find an apparently *detrimental* interaction among the full set of context features.

Theoretically adding new features should only improve a system as a feature can always by ignored by assigning it a weight of zero. However, new features expand the hypothesis space and provide the model with more degrees of freedom which may make it easier to get stuck in local minima. While the gradient-based, annealing method for optimization that we use tends work better than MERT (Och, 2003), it is still susceptible to these issues. Indeed, the variation on the tuning set–while relatively inconsequential–is evidence that this is occurring and that we have not found the global optimum. Further investigation is necessary into the interaction between the context features and techniques for robust optimization.

[3]These results have been updated since the official WMT '10 submission as a result of minor bug-fixes and code improvements to Cunei.

	German	English	Czech	English
Tokens	41,245,188	43,064,069	63,776,164	72,325,831
Sentences	1574044		6181270	

Table 1: Corpus Statistics

2.4 Conclusion

We used the Cunei Machine Translation Platform to build German to English and Czech to English systems for the WMT '10 evaluation. In both systems we experimented with per-instance alignment and context features. Our addition of the context features resulted in only minor improvement, but a deeper analysis of the individual features suggests greater potential. Overall, Cunei performed strongly in our evaluation against a comparable Moses system. We acknowledge that the actual features we selected are not particularly novel. Instead, the importance of this work is the simplicity with which instance-specific features can be jointly modeled and integrated within Cunei as a result of its unique modeling approach.

Acknowledgements

The author would like to thank Ralf Brown for providing suggestions and feedback on this paper.

References

Ondřej Bojar and Zdeněk Žabokrtský. 2009. Czeng0.9: Large parallel treebank with rich annotation. *Prague Bulletin of Mathematical Linguistics*, 92.

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. 2007. 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.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X Proceedings* (mts, 2005), pages 79–86.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. 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.

Percy Liang, Ben Taskar, and Dan Klein. 2006. 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.

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

2005. Phuket, Thailand, September.

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

Franz Josef Och. 2003. 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.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA, July.

Robert Parker et al. 2009. English gigaword fourth edition.

Aaron B. Phillips and Ralf D. Brown. 2009. Cunei machine translation platform: System description. In Mikel L. Forcada and Andy Way, editors, *Proceedings of the 3rd Workshop on Example-Based Machine Translation*, pages 29–36, Dublin, Ireland, November.

David A. Smith and Jason Eisner. 2006. 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.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing*, pages 901–904, Denver, USA, September.

Stephan Vogel. 2005. Pesa: Phrase pair extraction as sentence splitting. In *Machine Translation Summit X Proceedings* (mts, 2005), pages 251–258.

	Development Tuning				Development Test				Blind Test			
	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER
Moses	0.1916	5.9156	0.5286	0.6475	0.2046	6.2802	0.5330	0.6523	0.2097	6.5657	0.5591	0.6313
Cunei with Alignment	0.2018	5.9847	0.5326	0.6375	0.2125	6.3639	0.5342	0.6430	0.2210	6.6355	0.5573	0.6224
Cunei with Alignment & Context	0.2022	6.0021	0.5331	0.6362	0.2127	6.3753	0.5344	0.6408	0.2214	6.6467	0.5575	0.6198

Table 2: Overview of German to English Evaluations

	Development Tuning				Development Test				Blind Test			
	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER
Moses	0.2141	6.1969	0.5536	0.6170	0.2041	6.3574	0.5361	0.6422	0.2297	6.7916	0.5617	0.6054
Cunei with Alignment	0.2206	6.2634	0.5555	0.6128	0.2058	6.4116	0.5425	0.6391	0.2291	6.8464	0.5665	0.6003
Cunei with Alignment & Context	0.2170	6.2802	0.5567	0.6125	0.2065	6.4391	0.5398	0.6362	0.2315	6.8829	0.5676	0.5984

Table 3: Overview of Czech to English Evaluations

	Development Tuning				Development Test				Blind Test			
	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER	BLEU	NIST	Meteor	TER
Cunei	0.2018	5.9847	0.5326	0.6375	0.2125	6.3639	0.5342	0.6430	0.2210	6.6355	0.5573	0.6224
+ Origins	0.2010	6.0233	0.5370	0.6353	0.2150	6.4154	0.5361	0.6391	0.2221	6.6719	0.5609	0.6208
+ Adjacent Length & Skew	0.2002	6.0080	0.5338	0.6402	0.2147	6.4183	0.5354	0.6431	0.2237	6.7336	0.5574	0.6172
+ Adjacent N-grams	0.2011	5.9648	0.5310	0.6410	0.2137	6.3598	0.5329	0.6434	0.2235	6.6656	0.5564	0.6202
+ Doc Cosine & JSD	0.1987	5.9514	0.5305	0.6422	0.2134	6.3498	0.5324	0.6456	0.2228	6.6647	0.5579	0.6209
+ Doc Precision & Recall	0.2007	5.9764	0.5315	0.6376	0.2145	6.3984	0.5361	0.6410	0.2244	6.6900	0.5608	0.6206

Table 4: Breakdown of Context Features in German to English