Tunable Distortion Limits and Corpus Cleaning for SMT

Sara Stymne Christian Hardmeier Jörg Tiedemann Joakim Nivre
Uppsala University
Department of Linguistics and Philology
firstname.lastname@lingfil.uu.se

Abstract

We describe the Uppsala University system for WMT13, for English-to-German translation. We use the Docent decoder, a local search decoder that translates at the document level. We add tunable distortion limits, that is, soft constraints on the maximum distortion allowed, to Docent. We also investigate cleaning of the noisy Common Crawl corpus. We show that we can use alignment-based filtering for cleaning with good results. Finally we investigate effects of corpus selection for recasing.

1 Introduction

In this paper we present the Uppsala University submission to WMT 2013. We have submitted one system, for translation from English to German. In our submission we use the document-level decoder Docent (Hardmeier et al., 2012; Hardmeier et al., 2013). In the current setup, we take advantage of Docent in that we introduce *tunable distortion limits*, that is, modeling distortion limits as soft constraints instead of as hard constraints. In addition we perform experiments on corpus cleaning. We investigate how the noisy Common Crawl corpus can be cleaned, and suggest an *alignment-based* cleaning method, which works well. We also investigate corpus selection for recasing.

In Section 2 we introduce our decoder, Docent, followed by a general system description in Section 3. In Section 4 we describe our experiments with corpus cleaning, and in Section 5 we describe experiments with tunable distortion limits. In Section 6 we investigate corpus selection for recasing. In Section 7 we compare our results with Docent to results using Moses (Koehn et al., 2007). We conclude in Section 8.

2 The Docent Decoder

Docent (Hardmeier et al., 2013) is a decoder for phrase-based SMT (Koehn et al., 2003). It differs from other publicly available decoders by its use of a different search algorithm that imposes fewer restrictions on the feature models that can be implemented.

The most popular decoding algorithm for phrase-based SMT is the one described by Koehn et al. (2003), which has become known as *stack decoding*. It constructs output sentences bit by bit by appending phrase translations to an initially empty hypothesis. Complexity is kept in check, on the one hand, by a beam search approach that only expands the most promising hypotheses. On the other hand, a dynamic programming technique called *hypothesis recombination* exploits the locality of the standard feature models, in particular the n-gram language model, to achieve a loss-free reduction of the search space. While this decoding approach delivers excellent search performance at a very reasonable speed, it limits the information available to the feature models to an n-gram window similar to a language model history. In stack decoding, it is difficult to implement models with sentence-internal long-range dependencies and cross-sentence dependencies, where the model score of a given sentence depends on the translations generated for another sentence.

In contrast to this very popular stack decoding approach, our decoder Docent implements a search procedure based on *local search* (Hardmeier et al., 2012). At any stage of the search process, its search state consists of a complete document translation, making it easy for feature models to access the complete document with its current translation at any point in time. The search algorithm is a stochastic variant of standard *hill climbing*. At each step, it generates a successor of the current search state by randomly applying

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 225–231,
Sofia, Bulgaria, August 8-9, 2013

one of a set of state changing operations to a random location in the document. If the new state has a better score than the previous one, it is accepted, else search continues from the previous state. The operations are designed in such a way that every state in the search space can be reached from every other state through a sequence of state operations. In the standard setup we use three operations: *change-phrase-translation* replaces the translation of a single phrase with another option from the phrase table, *resegment* alters the phrase segmentation of a sequence of phrases, and *swap-phrases* alters the output word order by exchanging two phrases.

In contrast to stack decoding, the search algorithm in Docent leaves model developers much greater freedom in the design of their feature functions because it gives them access to the translation of the complete document. On the downside, there is an increased risk of search errors because the document-level hill-climbing decoder cannot make as strong assumptions about the problem structure as the stack decoder does. In practice, this drawback can be mitigated by initializing the hill-climber with the output of a stack decoding pass using the baseline set of models without document-level features (Hardmeier et al., 2012). Since its inception, Docent has been used to experiment with document-level semantic language models (Hardmeier et al., 2012) and models to enhance text readability (Stymne et al., 2013b). Work on other discourse phenomena is ongoing. In the present paper, we focus on sentence-internal reordering by exploiting the fact that Docent implements distortion limits as soft constraints rather than strictly enforced limitations. We do not include any of our document-level feature functions.

3 System Setup

In this section we will describe our basic system setup. We used all corpora made available for English–German by the WMT13 workshop. We always concatenated the two bilingual corpora Europarl and News Commentary, which we will call EP-NC. We pre-processed all corpora by using the tools provided for tokenization and we also lower-cased all corpora. For the bilingual corpora we also filtered sentence pairs with a length ratio larger than three, or where either sentence was longer than 60 tokens. Recasing was performed as a post-processing step, trained using the resources in the Moses toolkit (Koehn et al., 2007).

For the language model we trained two separate models, one on the German side of EP-NC, and one on the monolingual News corpus. In both cases we trained 5-gram models. For the large News corpus we used entropy-based pruning, with 10^{-8} as a threshold (Stolcke, 1998). The language models were trained using the SRILM toolkit (Stolcke, 2002) and during decoding we used the KenLM toolkit (Heafield, 2011).

For the translation model we also trained two models, one with EP-NC, and one with Common Crawl. These two models were interpolated and used as a single model at decoding time, based on perplexity minimization interpolation (Sennrich, 2012), see details in Section 4. The translation models were trained using the Moses toolkit (Koehn et al., 2007), with standard settings with 5 features, phrase probabilities and lexical weighting in both directions and a phrase penalty. We applied significance-based filtering (Johnson et al., 2007) to the resulting phrase tables. For decoding we used the Docent decoder with random initialization and standard parameter settings (Hardmeier et al., 2012; Hardmeier et al., 2013), which beside translation and language model features include a word penalty and a distortion penalty.

Parameter optimization was performed using MERT (Och, 2003) at the document-level (Stymne et al., 2013a). In this setup we calculate both model and metric scores on the document-level instead of on the sentence-level. We produce k-best lists by sampling from the decoder. In each optimization run we run 40,000 hill-climbing iterations of the decoder, and sample translations with interval 100, from iteration 10,000. This procedure has been shown to give competitive results to standard tuning with Moses (Koehn et al., 2007) with relatively stable results (Stymne et al., 2013a). For tuning data we concatenated the tuning sets news-test 2008–2010 and newssyscomb2009, to get a higher number of documents. In this set there are 319 documents and 7434 sentences.

To evaluate our system we use newstest2012, which has 99 documents and 3003 sentences. In this article we give lower-case Bleu scores (Papineni et al., 2002), except in Section 6 where we investigate the effect of different recasing models.

Cleaning	Sentences	Reduction
None	2,399,123	
Basic	2,271,912	5.3%
Langid	2,072,294	8.8%
Alignment-based	1,512,401	27.0%

Table 1: Size of Common Crawl after the different cleaning steps and reduction in size compared to the previous step

4 Cleaning of Common Crawl

The Common Crawl (CC) corpus was collected from web sources, and was made available for the WMT13 workshop. It is noisy, with many sentences with the wrong language and also many non-corresponding sentence pairs. To make better use of this resource we investigated two methods for cleaning it, by making use of language identification and alignment-based filtering. Before any other cleaning we performed basic filtering where we only kept pairs where both sentences had at most 60 words, and with a length ratio of maximum 3. This led to a 5.3% reduction of sentences, as shown in Table 1.

Language Identification For language identification we used the off-the-shelf tool langid.py (Lui and Baldwin, 2012). It is a python library, covering 97 languages, including English and German, trained on data drawn from five different domains. It uses a naive Bayes classifier with a multinomial event model, over a mixture of byte n-grams. As for many language identification packages it works best for longer texts, but Lui and Baldwin (2012) also showed that it has good performance for short microblog texts, with an accuracy of 0.89–0.94.

We applied langid.py for each sentence in the CC corpus, and kept only those sentence pairs where the correct language was identified for both sentences with a confidence of at least 0.999. The total number of sentences was reduced by a further 8.8% based on the langid filtering.

We performed an analysis on a set of 1000 sentence pairs. Among the 907 sentences that were kept in this set we did not find any cases with the wrong language. Table 2 shows an analysis of the 93 sentences that were removed from this test set. The overall accuracy of langid.py is much higher than indicated in the table, however, since it does not include the correctly identified English and German sentences. We grouped the removed sentences into four categories, cases where both languages were correctly identified, but under the confidence threshold of 0.999, cases where both languages were incorrectly identified, and cases where one language was incorrectly identified. Overall the language identification was accurate on 54 of the 93 removed sentences. In 18 of the cases where it was wrong, the sentences were not translation correspondents, which means that we only wrongly removed 21 out of 1000 sentences. It was also often the case when the language was wrongly identified, that large parts of the sentence consisted of place names, such as "Forums about Conil de la Frontera - Cádiz." – "Foren über Conil de la Frontera - Cádiz.", which were identified as es/ht instead of en/de. Even though such sentence pairs do correspond, they do not contain much useful translation material.

Alignment-Based Cleaning For the alignment-based cleaning, we aligned the data from the previous step using GIZA++ (Och and Ney, 2003) in both directions, and used the intersection of the alignments. The intersection of alignments is more sparse than the standard SMT symmetrization heuristics, like grow-diag-final-and (Koehn et al., 2005). Our hypothesis was that sentence pairs with very few alignment points in the intersection would likely not be corresponding sentences.

We used two types of filtering thresholds based on alignment points. The first threshold is for the ratio of the number of alignment points and the maximum sentence length. The second threshold is the absolute number of alignment points in a sentence pair. In addition we used a third threshold based on the length ratio of the sentences.

To find good values for the filtering thresholds, we created a small gold standard where we manually annotated 100 sentence pairs as being corresponding or not. In this set the sentence pairs did not match in 33 cases. Table 3 show results for some different values for the threshold parameters. Overall we are able to get a very high precision on the task of removing non-corresponding sentences, which means that most sentences that are removed based on this cleaning are actually non-corresponding sentences. The recall is a bit lower, indicating that there are still non-corresponding sentences left in our data. In our translation system we used the bold values in Table 3, since it gave high precision with reasonable recall for the removal of non-corresponding sentences, meaning

Identification	Total	Wrong lang.	Non-corr	Corr	Languages identified
English and German < 0.999	15	0	7	8	
Both English and German wrong	6	2	2	2	2:na/es, 2:et/et, 1: es/an, 1:es/ht
English wrong	13	1	6	6	5: es 4: fr 1: br, it, de, eo
German wrong	59	51	3	5	51: en 3: es 2:nl 1: af, la, lb
Total	93	54	18	21	

Table 2: Reasons and correctness for removing sentences based on language ID for 93 sentences out of a 1000 sentence subset, divided into wrong lang(uage), non-corr(esponding) pairs, and corr(esponding) pairs.

Ratio align	Min align	Ratio length	Prec.	Recall	F	Kept
0.1	4	2	0.70	0.77	0.73	70%
0.28	**4**	**2**	**0.94**	**0.72**	**0.82**	**57%**
0.42	4	2	1.00	0.56	0.72	41%
0.28	2	2	0.91	0.73	0.81	59%
0.28	6	2	0.94	0.63	0.76	51%
0.28	4	1.5	0.94	0.65	0.77	52%
0.28	4	3	0.91	0.75	0.82	60%

Table 3: Results of alignment-based cleaning for different values of the filtering parameters, with precision, recall and F-score for the identification of erroneous sentence pairs and the percentage of kept sentence pairs

that we kept most correctly aligned sentence pairs.

This cleaning method is more aggressive than the other cleaning methods we described. For the gold standard only 57% of sentences were kept, but in the full training set it was a bit higher, 73%, as shown in Table 1.

Phrase Table Interpolation To use the CC corpus in our system we first trained a separate phrase table which we then interpolated with the phrase table trained on EP-NC. In this way we could always run the system with a single phrase table. For interpolation, we used the perplexity minimization for weighted counts method by Sennrich (2012). Each of the four weights in the phrase table, backward and forward phrase translation probabilities and lexical weights, are optimized separately. This method minimizes the cross-entropy based on a held-out corpus, for which we used the concatenation of all available News development sets.

The cross-entropy and the contribution of CC relative to EP-NC, are shown for phrase translation probabilities in both directions in Table 4. The numbers for lexical weights show similar trends. For each cleaning step the cross-entropy is reduced and the contribution of CC is increased. The difference between the basic cleaning and langid is very small, however. The alignment-based cleaning shows a much larger effect. After that cleaning step the CC corpus has a similar contribution to EP-NC. This is an indicator that the final cleaned CC corpus fits the development set well.

	$p(S\|T)$		$p(T\|S)$	
Cleaning	CE	IP	CE	IP
Basic	3.18	0.12	3.31	0.06
Langid	3.17	0.13	3.29	0.07
Alignment-based	3.02	0.47	3.17	0.61

Table 4: Cross-entropy (CE) and relative interpolation weights (IP) compared to EP-NC for the Common Crawl corpus, with different cleaning

Results In Table 5 we show the translation results with the different types of cleaning of CC, and without it. We show results of different corpus combinations both during tuning and testing. We see that we get the overall best result by both tuning and testing with the alignment-based cleaning of CC, but it is not as useful to do the extra cleaning if we do not tune with it as well. Overall we get the best results when tuning is performed including a cleaned version of CC. This setup gives a large improvement compared to not using CC at all, or to use it with only basic cleaning. There is little difference in Bleu scores when testing with either basic cleaning, or cleaning based on language ID, with a given tuning, which is not surprising given their small and similar interpolation weights. Tuning was, however, not successful when using CC with basic cleaning.

Overall we think that alignment-based corpus cleaning worked well. It reduced the size of the corpus by over 25%, improved the cross-entropy for interpolation with the EP-NC phrase-table, and

	Testing			
Tuning	not used	basic	langid	alignment
not used	14.0	13.9	13.9	14.0
basic	14.2	14.5	14.3	14.3
langid	15.2	15.3	15.3	15.3
alignment	12.7	15.3	15.3	15.7

Table 5: Bleu scores with different types of cleaning and without Common Crawl

gave an improvement on the translation task. We still think that there is potential for further improving this filtering and to annotate larger test sets to investigate the effects in more detail.

5 Tunable Distortion Limits

The Docent decoder uses a hill-climbing search and can perform operations anywhere in the sentence. Thus, it does not need to enforce a strict distortion limit. In the Docent implementation, the distortion limit is actually implemented as a feature, which is normally given a very large weight, which effectively means that it works as a hard constraint. This could easily be relaxed, however, and in this work we investigate the effects of using soft distortion limits, which can be optimized during tuning, like other features. In this way long-distance movements can be allowed when they are useful, instead of prohibiting them completely. A drawback of using no or soft distortion limits is that it increases the search space.

In this work we mostly experiment with variants of one or two standard distortion limits, but with a tunable weight. We also tried to use separate soft distortion limits for left- and right-movement. Table 6 show the results with different types of distortion limits. The system with a standard fixed distortion limits of 6 has a somewhat lower score than most of the systems with no or soft distortion limits. In most cases the scores are similar, and we see no clear affects of allowing tunable limits over allowing unlimited distortion. The system that uses two mono-directional limits of 6 and 10 has slightly higher scores than the other systems, and is used in our final submission.

One possible reason for the lack of effect of allowing more distortion could be that it rarely happens that an operator is chosen that performs such distortion, when we use the standard Docent settings. To investigate this, we varied the settings of the parameters that guide the *swap-phrases* operator, and used the *move-phrases* operator instead of *swap-phrases*. None of these changes led to any improvements, however.

DL type	Limit	Bleu
No DL	–	15.5
Hard DL	6	15.0
One soft DL	6	15.5
	8	14.2
	10	15.5
Two soft DLs	4,8	15.5
	6,10	15.7
Bidirectional soft DLs	6,10	15.5

Table 6: Bleu scores for different distortion limit (DL) settings

While we saw no clear effects when using tunable distortion limits, we plan to extend this work in the future to model movement differently based on parts of speech. For the English–German language pair, for instance, it would be reasonable to allow long distance moves of verb groups with no or little cost, but use a hard limit or a high cost for other parts of speech.

6 Corpus Selection for Recasing

In this section we investigate the effect of using different corpus combinations for recasing. We lower-cased our training corpus, which means that we need a full recasing step as post-processing. This is performed by training a SMT system on lower-cased and true-cased target language. We used the Moses toolkit to train the recasing system and to decode during recasing. We investigate the effect of using different combinations of the available training corpora to train the recasing model.

Table 7 show case sensitive Bleu scores, which can be compared to the previous case-insensitive scores of 15.7. We see that there is a larger effect of including more data in the language model than in the translation model. There is a performance jump both when adding CC data and when adding News data to the language model. The results are best when we include the News data, which is not included in the English–German translation model, but which is much larger than the other corpora. There is no further gain by using News in combination with other corpora compared to using only News. When adding more data to the translation model there is only a minor effect, with the difference between only using EP-NC and using all available corpora is at most 0.2 Bleu points. In our submitted system we use the monolingual News corpus both in the LM and the TM.

There are other options for how to treat recas-

	Language model				
TM	EP-NC	EP-NC-CC	News	EP-NC-News	EP-NC-CC-News
EP-NC	13.8	14.4	14.8	14.8	14.8
EP-NC-CC	13.9	14.5	14.9	14.8	14.8
News	13.9	14.5	14.9	14.9	14.9
EP-NC-News	13.9	14.5	14.9	14.9	14.9
EP-NC-CC-News	13.9	14.5	14.9	14.9	15.0

Table 7: Case-sensitive Bleu scores with different corpus combinations for the language model and translation model (TM) for recasing

ing. It is common to train the system on true-cased data instead of lower-cased data, which has been shown to lead to small gains for the English–German language pair (Koehn et al., 2008). In this framework there is still a need to find the correct case for the first word of each sentence, for which a similar corpus study might be useful.

7 Comparison to Moses

So far we have only shown results using the Docent decoder on its own, with a random initialization, since we wanted to submit a Docent-only system for the shared task. In this section we also show contrastive results with Moses, and for Docent initialized with stack decoding, using Moses, and for different type of tuning.

Previous research have shown mixed results for the effect of initializing Docent with and without stack decoding, when using the same feature sets. In Hardmeier et al. (2012) there was a drop of about 1 Bleu point for English–French translation based on WMT11 data when random initialization was used. In Stymne et al. (2013a), on the other hand, Docent gave very similar results with both types of initialization for German–English WMT13 data. The latter setup is similar to ours, except that no Common Crawl data was used.

The results with our setup are shown in Table 8. In this case we lose around a Bleu point when using Docent on its own, without Moses initialization. We also see that the results are lower when using Moses with the Docent tuning method, or when combining Moses and Docent with Docent tuning. This indicates that the document-level tuning has not given satisfactory results in this scenario, contrary to the results in Stymne et al. (2013a), which we plan to explore further in future work. Overall we think it is important to develop stronger context-sensitive models for Docent, which can take advantage of the document context.

Test system	Tuning system	Bleu
Docent (random)	Docent	15.7
Docent (stack)	Docent	15.9
Moses	Docent	15.9
Docent (random)	Moses	15.9
Docent (stack)	Moses	16.8
Moses	Moses	16.8

Table 8: Bleu scores for Docent initialized randomly or with stack decoding compared to Moses. Tuning is performed with either Moses or Docent. For the top line we used tunable distortion limits 6,10 with Docent, in the other cases a standard hard distortion limit of 6, since Moses does not allow soft distortion limits.

8 Conclusion

We have presented the Uppsala University system for WMT 2013. Our submitted system uses Docent with random initialization and two tunable distortion limits of 6 and 10. It is trained with the Common Crawl corpus, cleaned using language identification and alignment-based filtering. For recasing we used the monolingual News corpora.

For corpus-cleaning, we present a novel method for cleaning noisy corpora based on the number and ratio of word alignment links for sentence pairs, which leads to a large reduction of corpus size, and to small improvements on the translation task. We also experiment with tunable distortion limits, which do not lead to any consistent improvements at this stage.

In the current setup the search algorithm of Docent is not strong enough to compete with the effective search in standard decoders like Moses. We are, however, working on developing discourse-aware models that can take advantage of the document-level context, which is available in Docent. We also need to further investigate tuning methods for Docent.

References

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea.

Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A document-level decoder for phrase-based statistical machine translation. In *Proceedings of the 51st Annual Meeting of the ACL, Demonstration session*, Sofia, Bulgaria.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 967–975, Prague, Czech Republic.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the NAACL*, pages 48–54, Edmonton, Alberta, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

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 ACL, Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Philipp Koehn, Abhishek Arun, and Hieu Hoang. 2008. Towards better machine translation quality for the German-English language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 139–142, Columbus, Ohio, USA.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the 50th Annual Meeting of the ACL, System Demonstrations*, pages 25–30, Jeju Island, Korea.

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 42nd Annual Meeting of the ACL*, pages 160–167, Sapporo, Japan.

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 ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 539–549, Avignon, France.

Andreas Stolcke. 1998. Entropy-based pruning of backoff language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274, Landsdowne, Virginia, USA.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA.

Sara Stymne, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2013a. Feature weight optimization for discourse-level SMT. In *Proceedings of the ACL 2013 Workshop on Discourse in Machine Translation (DiscoMT 2013)*, Sofia, Bulgaria.

Sara Stymne, Jörg Tiedemann, Christian Hardmeier, and Joakim Nivre. 2013b. Statistical machine translation with readability constraints. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NODALIDA'13)*, pages 375–386, Oslo, Norway.