Docent: A Document-Level Decoder for Phrase-Based Statistical Machine Translation

Christian Hardmeier Sara Stymne Jörg Tiedemann Joakim Nivre
Uppsala University
Department of Linguistics and Philology
Box 635, 751 26 Uppsala, Sweden
firstname.lastname@lingfil.uu.se

Abstract

We describe Docent, an open-source decoder for statistical machine translation that breaks with the usual sentence-by-sentence paradigm and translates complete documents as units. By taking translation to the document level, our decoder can handle feature models with arbitrary discourse-wide dependencies and constitutes an essential infrastructure component in the quest for discourse-aware SMT models.

1 Motivation

Most of the research on statistical machine translation (SMT) that was conducted during the last 20 years treated every text as a "bag of sentences" and disregarded all relations between elements in different sentences. Systematic research into explicitly discourse-related problems has only begun very recently in the SMT community (Hardmeier, 2012) with work on topics such as pronominal anaphora (Le Nagard and Koehn, 2010; Hardmeier and Federico, 2010; Guillou, 2012), verb tense (Gong et al., 2012) and discourse connectives (Meyer et al., 2012).

One of the problems that hamper the development of cross-sentence models for SMT is the fact that the assumption of sentence independence is at the heart of the dynamic programming (DP) beam search algorithm most commonly used for decoding in phrase-based SMT systems (Koehn et al., 2003). For integrating cross-sentence features into the decoding process, researchers had to adopt strategies like two-pass decoding (Le Nagard and Koehn, 2010). We have previously proposed an algorithm for document-level phrase-based SMT decoding (Hardmeier et al., 2012). Our decoding algorithm is based on local search instead of dynamic programming and permits the integration of document-level models with unrestricted dependencies, so that a model score can be conditioned on arbitrary elements occurring anywhere in the input document or in the translation that is being generated. In this paper, we present an open-source implementation of this search algorithm. The decoder is written in C++ and follows an object-oriented design that makes it easy to extend it with new feature models, new search operations or different types of local search algorithms. The code is released under the GNU General Public License and published on Github[1] to make it easy for other researchers to use it in their own experiments.

2 Document-Level Decoding with Local Search

Our decoder is based on the phrase-based SMT model described by Koehn et al. (2003) and implemented, for example, in the popular Moses decoder (Koehn et al., 2007). Translation is performed by splitting the input sentence into a number of contiguous word sequences, called phrases, which are translated into the target language through a phrase dictionary lookup and optionally reordered. The choice between different translations of an ambiguous source phrase and the ordering of the target phrases are guided by a scoring function that combines a set of scores taken from the phrase table with scores from other models such as an n-gram language model. The actual translation process is realised as a search for the highest-scoring translation in the space of all the possible translations that could be generated given the models.

The decoding approach that is implemented in Docent was first proposed by Hardmeier et al. (2012) and is based on local search. This means that it has a state corresponding to a complete, if possibly bad, translation of a document at every

[1] https://github.com/chardmeier/docent/wiki

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 193–198,
Sofia, Bulgaria, August 4-9 2013.

stage of the search progress. Search proceeds by making small changes to the current search state in order to transform it gradually into a better translation. This differs from the DP algorithm used in other decoders, which starts with an empty translation and expands it bit by bit. It is similar to previous work on phrase-based SMT decoding by Langlais et al. (2007), but enables the creation of document-level models, which was not addressed by earlier approaches.

Docent currently implements two search algorithms that are different generalisations of the hill climbing local search algorithm by Hardmeier et al. (2012). The original hill climbing algorithm starts with an initial state and generates possible successor states by randomly applying simple elementary operations to the state. After each operation, the new state is scored and accepted if its score is better than that of the previous state, else rejected. Search terminates when the decoder cannot find an acceptable successor state after a certain number of attempts, or when a maximum number of steps is reached.

Simulated annealing is a stochastic variant of hill climbing that always accepts moves towards better states, but can also accept moves towards lower-scoring states with a certain probability that depends on a temperature parameter in order to escape local maxima. *Local beam search* generalises hill climbing in a different way by keeping a beam of a fixed number of multiple states at any time and randomly picking a state from the beam to modify at each move. The original hill climbing procedure can be recovered as a special case of either one of these search algorithms, by calling simulated annealing with a fixed temperature of 0 or local beam search with a beam size of 1.

Initial states for the search process can be generated either by selecting a random segmentation with random translations from the phrase table in monotonic order, or by running DP beam search with sentence-local models as a first pass. For the second option, which generally yields better search results, Docent is linked with the Moses decoder and makes direct calls to the DP beam search algorithm implemented by Moses. In addition to these state initialisation procedures, Docent can save a search state to a disk file which can be loaded again in a subsequent decoding pass. This saves time especially when running repeated experiments from the same starting point obtained by DP search.

In order to explore the complete search space of phrase-based SMT, the search operations in a local search decoder must be able to change the phrase translations, the order of the output phrases and the segmentation of the source sentence into phrases. The three operations used by Hardmeier et al. (2012), *change-phrase-translation*, *resegment* and *swap-phrases*, jointly meet this requirement and are all implemented in Docent. Additionally, Docent features three extra operations, all of which affect the target word order: The *move-phrases* operation moves a phrase to another location in the sentence. Unlike *swap-phrases*, it does not require that another phrase be moved in the opposite direction at the same time. A pair of operations called *permute-phrases* and *linearise-phrases* can reorder a sequence of phrases into random order and back into the order corresponding to the source language.

Since the search algorithm in Docent is stochastic, repeated runs of the decoder will generally produce different output. However, the variance of the output is usually small, especially when initialising with a DP search pass, and it tends to be lower than the variance introduced by feature weight tuning (Hardmeier et al., 2012; Stymne et al., 2013a).

3 Available Feature Models

In its current version, Docent implements a selection of sentence-local feature models that makes it possible to build a baseline system with a configuration comparable to that of a typical Moses baseline system. The published source code also includes prototype implementations of a few document-level models. These models should be considered work in progress and serve as a demonstration of the cross-sentence modelling capabilities of the decoder. They have not yet reached a state of maturity that would make them suitable for production use.

The sentence-level models provided by Docent include the phrase table, n-gram language models implemented with the KenLM toolkit (Heafield, 2011), an unlexicalised distortion cost model with geometric decay (Koehn et al., 2003) and a word penalty cost. All of these features are designed to be compatible with the corresponding features in Moses. From among the typical set of baseline features in Moses, we have not implemented the

lexicalised distortion model, but this model could easily be added if required. Docent uses the same binary file format for phrase tables as Moses, so the same training apparatus can be used.

DP-based SMT decoders have a parameter called distortion limit that limits the difference in word order between the input and the MT output. In DP search, this is formally considered to be a parameter of the search algorithm because it affects the algorithmic complexity of the search by controlling how many translation options must be considered at each hypothesis expansion. The stochastic search algorithm in Docent does not require this limitation, but it can still be useful because the standard models of SMT do not model long-distance reordering well. Docent therefore includes a separate indicator feature to indicate a violated distortion limit. In conjunction with a very large weight, this feature can effectively ensure that the distortion limit is enforced. In contrast with the distortion limit parameter of a DP decoder, the weight of our distortion limit feature can potentially be tuned to permit occasional distortion limit violations when they contribute to better translations.

The document-level models included in Docent include a *length parity model*, a *semantic language model* as well as a collection of document-level *readability models*. The length parity model is a proof-of-concept model that ensures that all sentences in a document have either consistently odd or consistently even length. It serves mostly as a template to demonstrate how a simple document-level model can be implemented in the decoder. The semantic language model was originally proposed by Hardmeier et al. (2012) to improve lexical cohesion in a document. It is a cross-sentence model over sequences of content words that are scored based on their similarity in a word vector space. The readability models serve to improve the readability of the translation by encouraging the selection of easier and more consistent target words. They are described and demonstrated in more detail in section 5.

Docent can read input files both in the NIST-XML format commonly used to encode documents in MT shared tasks such as NIST or WMT and in the more elaborate MMAX format (Müller and Strube, 2003). The MMAX format makes it possible to include a wide range of discourse-level corpus annotations such as coreference links. These annotations can then be accessed by the feature models. To allow for additional target-language information such as morphological features of target words, Docent can handle simple word-level annotations that are encoded in the phrase table in the same way as target language factors in Moses.

In order to optimise feature weights we have adapted the Moses tuning infrastructure to Docent. In this way we can take advantage of all its features, for instance using different optimisation algorithms such as MERT (Och, 2003) or PRO (Hopkins and May, 2011), and selective tuning of a subset of features. Since document features only give meaningful scores on the document level and not on the sentence level, we naturally perform optimisation on document level, which typically means that we need more data than for the optimisation of sentence-based decoding. The results we obtain are relatively stable and competitive with sentence-level optimisation of the same models (Stymne et al., 2013a).

4 Implementing Feature Models Efficiently

While translating a document, the local search decoder attempts to make a great number of moves. For each move, a score must be computed and tested against the acceptance criterion. An overwhelming majority of the proposed moves will be rejected. In order to achieve reasonably fast decoding times, efficient scoring is paramount. Recomputing the scores of the whole document at every step would be far too slow for the decoder to be useful. Fortunately, score computation can be sped up in two ways. Knowledge about how the state to be scored was generated from its predecessor helps to limit recomputations to a minimum, and by adopting a two-step scoring procedure that just computes the scores that can be calculated with little effort at first, we need to compute the complete score only if the new state has some chance of being accepted.

The scores of SMT feature models can usually be decomposed in some way over parts of the document. The traditional models borrowed from sentence-based decoding are necessarily decomposable at the sentence level, and in practice, all common models are designed to meet the constraints of DP beam search, which ensures that they can in fact be decomposed over even smal-

ler sequences of just a few words. For genuine document-level features, this is not the case, but even these models can often be decomposed in some way, for instance over paragraphs, anaphoric links or lexical chains. To take advantage of this fact, feature models in Docent always have access to the previous state and its score and to a list of the state modifications that transform the previous state into the next. The scores of the new state are calculated by identifying the parts of a document that are affected by the modifications, subtracting the old scores of this part from the previous score and adding the new scores. This approach to scoring makes feature model implementation a bit more complicated than in DP search, but it gives the feature models full control over how they decompose a document while still permitting efficient decoding.

A feature model class in Docent implements three methods. The *initDocument* method is called once per document when decoding starts. It straightforwardly computes the model score for the entire document from scratch. When a state is modified, the decoder first invokes the *estimateScoreUpdate* method. Rather than calculating the new score exactly, this method is only required to return an upper bound that reflects the maximum score that could possibly be achieved by this state. The search algorithm then checks this upper bound against the acceptance criterion. Only if the upper bound meets the criterion does it call the *updateScore* method to calculate the exact score, which is then checked against the acceptance criterion again.

The motivation for this two-step procedure is that some models can compute an upper bound approximation much more efficiently than an exact score. For any model whose score is a log probability, a value of 0 is a loose upper bound that can be returned instantly, but in many cases, we can do much better. In the case of the n-gram language model, for instance, a more accurate upper bound can be computed cheaply by subtracting from the old score all log-probabilities of n-grams that are affected by the state modifications without adding the scores of the n-grams replacing them in the new state. This approximation can be calculated without doing any language model lookups at all. On the other hand, some models like the distortion cost or the word penalty are very cheap to compute, so that the *estimateScoreUpdate* method can simply return the precise score as a tight upper bound. If a state gets rejected because of a low score on one of the cheap models, this means we will never have to compute the more expensive feature scores at all.

5 Readability: A Case Study

As a case study we report initial results on how document-wide features can be used in Docent in order to improve the readability of texts by encouraging simple and consistent terminology (Stymne et al., 2013b). This work is a first step towards achieving joint SMT and text simplification, with the final goal of adapting MT to user groups such as people with reading disabilities.

Lexical consistency modelling for SMT has been attempted before. The suggested approaches have been limited by the use of sentence-level decoders, however, and had to resort to procedures like post processing (Carpuat, 2009), multiple decoding runs with frozen counts from previous runs (Ture et al., 2012), or cache-based models (Tiedemann, 2010). In Docent, however, we always have access to a full document translation, which makes it straightforward to include features directly into the decoder.

We implemented four features on the document level. The first two features are type token ratio (TTR) and a reformulation of it, OVIX, which is less sensitive to text length. These ratios have been related to the "idea density" of a text (Mühlenbock and Kokkinakis, 2009). We also wanted to encourage consistent translations of words, for which we used the Q-value (Deléger et al., 2006), which has been proposed to measure term quality. We applied it on word level (QW) and phrase level (QP). These features need access to the full target document, which we have in Docent. In addition, we included two sentence-level count features for long words that have been used to measure the readability of Swedish texts (Mühlenbock and Kokkinakis, 2009).

We tested our features on English–Swedish translation using the Europarl corpus. For training we used 1,488,322 sentences. As test data, we extracted 20 documents with a total of 690 sentences. We used the standard set of baseline features: 5-gram language model, translation model with 5 weights, a word penalty and a distortion penalty.

Baseline	Readability features	Comment
de ärade ledamöterna (the honourable Members)	ledamöterna (the members) / ni (you)	+ Removal of non-essential words
på ett sådant sätt att (in such a way that)	så att (so that)	+ Simplified expression
gemenskapslagstiftningen (the community legislation)	gemenskapens lagstiftning (the community's legislation)	+ Shorter words by changing long compound to genitive construction
Världshandelsorganisationen (World Trade Organisation)	WTO (WTO)	− Changing long compound to English-based abbreviation
handlingsplanen (the action plan)	planen (the plan)	− Removal of important word
ägnat särskild uppmärksamhet åt (paid particular attention to)	särskilt uppmärksam på (particular attentive on)	− Bad grammar because of changed part of speech and missing verb

Table 2: Example translation snippets with comments

Feature	BLEU	OVIX	LIX
Baseline	0.243	56.88	51.17
TTR	0.243	55.25	51.04
OVIX	0.243	54.65	51.00
QW	0.242	57.16	51.16
QP	0.243	57.07	51.06
All	0.235	47.80	49.29

Table 1: Results for adding single lexical consistency features to Docent

To evaluate our system we used the BLEU score (Papineni et al., 2002) together with a set of readability metrics, since readability is what we hoped to improve by adding consistency features. Here we used OVIX to confirm a direct impact on consistency, and LIX (Björnsson, 1968), which is a common readability measure for Swedish. Unfortunately we do not have access to simplified translated text, so we calculate the MT metrics against a standard reference, which means that simple texts will likely have worse scores than complicated texts closer to the reference translation.

We tuned the standard features using Moses and MERT, and then added each lexical consistency feature with a small weight, using a grid search approach to find values with a small impact. The results are shown in Table 1. As can be seen, for individual features the translation quality was maintained, with small improvements in LIX, and in OVIX for the TTR and OVIX features. For the combination we lost a little bit on translation quality, but there was a larger effect on the readability metrics. When we used larger weights, there was a bigger impact on the readability metrics, with a further decrease on MT quality.

We also investigated what types of changes the readability features could lead to. Table 2 shows a sample of translations where the baseline is compared to systems with readability features. There are both cases where the readability features help and cases where they are problematic. Overall, these examples show that our simple features can help achieve some interesting simplifications.

There is still much work to do on how to take best advantage of the possibilities in Docent in order to achieve readable texts. This attempt shows the feasibility of the approach. We plan to extend this work for instance by better feature optimisation, by integrating part-of-speech tags into our features in order to focus on terms rather than common words, and by using simplified texts for evaluation and tuning.

6 Conclusions

In this paper, we have presented Docent, an open-source document-level decoder for phrase-based SMT released under the GNU General Public License. Docent is the first decoder that permits the inclusion of feature models with unrestricted dependencies between arbitrary parts of the output, even crossing sentence boundaries. A number of research groups have recently started to investigate the interplay between SMT and discourse-level phenomena such as pronominal anaphora, verb tense selection and the generation of discourse connectives. We expect that the availability of a document-level decoder will make it substantially easier to leverage discourse information in SMT and make SMT models explore new ground beyond the next sentence boundary.

References

Carl-Hugo Björnsson. 1968. *Läsbarhet*. Liber, Stockholm.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 19–27, Boulder, Colorado.

Louise Deléger, Magnus Merkel, and Pierre Zweigenbaum. 2006. Enriching medical terminologies: an approach based on aligned corpora. In *International Congress of the European Federation for Medical Informatics*, pages 747–752, Maastricht, The Netherlands.

Zhengxian Gong, Min Zhang, Chew Lim Tan, and Guodong Zhou. 2012. N-gram-based tense models for statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 276–285, Jeju Island, Korea.

Liane Guillou. 2012. Improving pronoun translation for statistical machine translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10, Avignon, France.

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 283–289, Paris, France.

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. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours*, 11.

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.

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

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*, pages 48–54, Edmonton.

Philipp Koehn, Hieu Hoang, Alexandra Birch, et al. 2007. Moses: open source toolkit for Statistical Machine Translation. In *Annual meeting of the Association for Computational Linguistics: Demonstration session*, pages 177–180, Prague, Czech Republic.

Philippe Langlais, Alexandre Patry, and Fabrizio Gotti. 2007. A greedy decoder for phrase-based statistical machine translation. In *TMI-2007: Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 104–113, Skövde, Sweden.

Ronan Le Nagard and Philipp Koehn. 2010. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261, Uppsala, Sweden.

Thomas Meyer, Andrei Popescu-Belis, Najeh Hajlaoui, and Andrea Gesmundo. 2012. Machine translation of labeled discourse connectives. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, San Diego, California, USA.

Katarina Mühlenbock and Sofie Johansson Kokkinakis. 2009. LIX 68 revisited – an extended readability. In *Proceedings of the Corpus Linguistics Conference*, Liverpool, UK.

Christoph Müller and Michael Strube. 2003. Multi-level annotation in MMAX. In *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, pages 198–207, Sapporo, Japan.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 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 Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Sara Stymne, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2013a. Feature weight optimization for discourse-level SMT. In *Proceedings of the Workshop on Discourse in Machine Translation (DiscoMT)*, 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 of Computational Linguistics (NODALIDA 2013)*, pages 375–386, Oslo, Norway.

Jörg Tiedemann. 2010. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the ACL 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP)*, pages 8–15, Uppsala, Sweden.

Ferhan Ture, Douglas W. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Montréal, Canada.