Two-Step Machine Translation with Lattices

Bushra Jawaid, Ondřej Bojar

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské nám. 25, Praha 1, CZ-118 00, Czech Republic
{jawaid,bojar}@ufal.mff.cuni.cz

Abstract

The idea of two-step machine translation was introduced to divide the complexity of the search space into two independent steps: (1) lexical translation and reordering, and (2) conjugation and declination in the target language. In this paper, we extend the two-step machine translation structure by replacing state-of-the-art phrase-based machine translation with the hierarchical machine translation in the 1^{st} step. We further extend the fixed string-based input format of the 2^{nd} step with word lattices (Dyer et al., 2008); this provides the 2^{nd} step with the opportunity to choose among a sample of possible reorderings instead of relying on the single best one as produced by the 1^{st} step.

Keywords: machine translation, morphologically rich language, word lattices

1. Introduction

The idea of decomposing the search for the best machine translation of a sentence into two steps to reduce the complexity of the problem is not new. For instance, Costa-jussà et al. (2007) use it for dealing with word order differences in source and target language. Minkov and Toutanova (2007) and Toutanova et al. (2008) are the first to use two step scheme to integrate the morpho-syntactic information to phrase-based machine translation systems. This approach mainly focuses on translating in the hard direction, that is translating from morphologically poor to morphologically richer languages.

The existing linguistically motivated shallow translation models such as factored-based models (Koehn and Hoang, 2007) work well in small data settings, or if the setup is kept very simple and almost preserves the search space of a simple phrase-based model based on word forms. The complex setups of factored models that are capable of the necessary generalizations are prohibitively expensive for large data, see also Bojar et al. (2009). In factored models, each token consists of a number of different factors such as surface form, lemma and so on. Translation options are constructed by mapping source factors to target factors. Depending on the exact translation setup, each translation option needs to be fully constructed before the actual search takes place. The more generalization we allow in the model, e.g. translating lemmas independently of morphological properties, the harder the search space explosion strikes (Bojar and Kos, 2010).

The idea behind using two-step machine translation is to avoid the explosion of the search space by dealing with reordering and word inflection in separate steps. In the first step, only morphological features common to both source and target are handled together with word reordering while target-specific morphological features are introduced in the second step only. This reduces the risk of the combinatorial explosion, because the target side of the first step is not cluttered with the information not available in the source language.

2. Common Settings

For the training of our two-step translation systems, we use Moses[1] (Koehn et al., 2007) along with the GIZA++ (Och and Ney, 2000) alignment toolkit and a 5-gram SRILM language model (Stolcke, 2002). The texts were processed using the Treex platform (Popel and Žabokrtský, 2010)[2], which included lemmatization and tagging by Morče (Spoustová et al., 2007).

Our training data is summarized in Table 1. To match the targeted setting, the translation model of the 1^{st} step is trained on small parallel data only. The translation model of the 2^{nd} step is trained on large monolingual data. Language models for both steps are built using the large monolingual data.

Dataset	Sents (cs/en)	Tokens (cs/en)	Source
Small	197k parallel	4.2M/4.8M	CzEng 1.0 news
Mono	18M/50M	317M/1.265G	WMT12 mono

Table 1: Summary of training data.

We use the official WMT11[3] test set for tuning and report final BLEU scores (Papineni et al., 2002) on the WMT12[4] official test set.

3. Two-Step Experiments

Our basic two-step translation scheme is similar to the two-step systems presented by Bojar et al. (2012). They used Moses in the first step to translate the source lemma or form to "augmented lemmatized output". This output of the first step is not a fully inflected target language. Instead, it represents an intermediate language consisting of word lemmas and a few morphological features. The second step is responsible for picking the values for the outstanding morphological features and constructing the word forms.

[1] http://statmt.org/moses/
[2] http://ufal.mff.cuni.cz/treex/
[3] http://www.statmt.org/wmt11/
[4] http://www.statmt.org/wmt12/

It is implemented as a monotone token-for-token translation where another Moses system is trained on augmented lemmatized sentences on the source side and their fully inflected versions on the target side, both coming from the monolingual corpus of the target language.

3.1. Intermediate Language Representation

We use the same representation for tokens in the intermediate language, i.e. the output of the 1^{st} and input of the 2^{nd} step which Bojar et al. (2012) use. For each token, we use LOF (lemma or form) for representing the lexical information and MOT (modified tag) for representing its morphological information. For most frequent words, LOF uses the full form of the word, otherwise it simply uses lemma.

We experiment with different MOTs where MOT_0 represents the most detailed morphological tag and $MOT_{1,2,...}$ represent modified tags we have created to reduce the size of the tagset and to omit some less important information.

MOT_1 uses a more coarse grained part of speech (POS) than MOT_0. Depending on the POS, different attributes are included: gender and number for nouns and pronouns; negation for nouns; person for pronouns; grade and negation for adjectives; case for prepositions and finally tense, negation and voice for verbs. The remaining grammatical categories are encoded using POS, number, grade and negation.

3.2. Experiment Configurations

Our baseline setup is a direct translation from the source form to the target form. In contrast to Bojar et al. (2012), we use two factors on the source side of the first step: not only the source word form but also the morphological tag. In small data settings, this is usually beneficial because it avoids the frequent noun-verb ambiguity in English. The rest of our baseline setup is similar.

We follow Bojar et al. (2012) formulation to represent the middle language: using LOF+MOT (single token) or LOF|MOT (multi-factor token). The first representation is simply created by concatenating the LOF and the MOT, applying a single language model to them, whereas in the later representation two separate LMs are used one for each factor. We keep the translation model practically identical in both variants: we use just one translation step, always producing both LOF and MOT at once. This approach obviously lacks some generalization but it keeps the search space simple.

The two step experiments are set up in two directions: either use single or multiple factors on the source side, and represent intermediate Czech either as a single token i.e. LOF+MOT or multi-factor token i.e. LOF|MOT.

As mentioned, the decoding path for both non-factored and factored setup consists of single translation step only, i.e. to translate form and tag to LOF|MOT in the first step and then LOF|MOT to the final form in the second step. In the compact notation of Bojar et al. (2012), this would be called tFaT-LOFaMOT = LOFaMOT-F. We also disregard the additional back-off decoding path (i.e. translating only from LOF to form) used by Bojar et al. (2012) in the 2^{nd} step.

Table 2 reports the BLEU scores when changing the number of factors ("+" vs. "|") in the middle language, the type of the LOF and MOT and the number of factors on the source side (Form only vs. Form and Tag).

MOTS	Source Factors		Factors in Middle Language +	Factors in Middle Language |
Baseline			13.23±0.49	
MOT_0	Form		12.70±0.52	12.13±0.44
	Form, Tag		12.71±0.49	12.26±0.46
MOT_1	Form	•	13.52±0.51	13.07±0.54
	Form, Tag	∗	13.46±0.53	13.09±0.53

Table 2: Two-step baseline experiments.

BLEU scores in Table 2 are not comparable to the results reported in Bojar et al. (2012). They used a language model build on small monolingual data for both steps and they also used small monolingual data to train the monotone translation model of the 2^{nd} step. We see 0.3 gain in BLEU score when using only form on the source side and single factor in the middle language i.e. LOF+MOT.

The results marked with • and ∗ were tested with MultEval[5] for statistical significance of the improvement over the baseline. Based on 3 independent MERT runs of both the baseline and the experiment in question, • marks the 99% confidence and ∗ marks the 91% confidence on the improvement over the baseline.

4. Hierarchical MT in the 1^{st} Step

Translation of language pairs having significant word order differences is a challenging task for the current state-of-the-art phrase-based MT systems. For instance, by default Moses provides "distance-based" reordering model that simply makes the translation more expensive if more words are skipped when taking source phrases out of their original sequence.

One of the more appropriate models for dealing with word reordering issues is the "hierarchical phrase-based models" (Chiang, 2005; Chiang, 2007). These models are formally built upon Synchronous Context-Free Grammar (SCFG) rules and they allow block movements which could help in improving reordering. In the Moses toolkit, they are implemented in the moses-chart executable.

MOTS	Source Factors		Factors in Middle Language +	Factors in Middle Language |
Baseline			13.43±0.53	
MOT_0	Form		12.84±0.51	12.52±0.48
	Form, Tag		12.85±0.52	12.67±0.48
MOT_1	Form	†	13.58±0.50	12.95±0.52
	Form, Tag	‡	13.56±0.50	13.07±0.51

Table 3: Two-step experiments using Hierarchical MT on 1st step

In Table 3, we list the BLEU scores after replacing the phrase-based model in the 1^{st} step with the hierarchical

[5] https://github.com/jhclark/multeval

one. We present the results of the same set of variations as in Table 2. These experiments are run with the default settings (stack size, max-phrase-length etc.) on 1^{st} step whereas on 2^{nd} step we set `-max-phrase-length` to 1 and `-distortion-limit` to 0 for monotone translation.

The experimental setting of taking LOF+MOT in the middle language and using form or form and tag on the source side brings a slight improvement over the hierarchical baseline score. The same setting showed an increase in BLEU score when we use phrase-based decoder in the 1^{st} step. Based on MultEval of three independent MERT runs, results marked with † and ‡ are significantly better than the hierarchical baseline at the confidence level of 95% and 92%, respectively.

5. Lattices in Middle Layer

Lattices as a form of compact representation of many possible analyses or annotations have been used to improve NLP tasks over a long period of time. Preserving ambiguity of the input or in an internal stage is often beneficial because the final processing step gets a chance to recover from errors in earlier stages.

The use of lattices in Machine Translation (MT) is not novel. They have already been used in various ways such as for overcoming the uncertainty about word boundaries in Chinese (Xu et al., 2005), the decoder is fed with an input lattice containing multiple segmentations of the source sentence. Lattices allow us to introduce systematic ambiguity in the input giving decoder freedom to pick the best input variation.

Deneefe et al. (2008) overcome the vocabulary sparseness in Arabic-English MT by using input word lattices of multiple source analyses of test sentences. The use of lattice in our work is more similar to the approach used by Costa-jussà et al. (2007); their SMR (Statistical Machine Reordering) model is split into two steps where the SMT system in the first step performs the translation from source S to S', producing weighted output word graphs consisting of multiple reorderings of source sentences. The system on the second step takes the weighted reordering graphs and produces final translations.

In all the experimental settings so far, our systems in the first step always produced the 1-best reordering for each sentence. We extend the string-based output of the first step to the form of a *word lattice* (Dyer et al., 2008). In other words, the middle language will now represent multiple reorderings for each source sentence, allowing the second step to choose among the reordered sentences the one that is the easiest to inflect.

For lattice experiments, only phrase-based MT is used on both steps and we use only experimental setup of MOT_1, i.e. one of the setups that was significantly better than the baseline.

Table 4 summarizes all results of these experiments and we now walk over the various setups we examined.

5.1. Lattice Pruning

The full search graph of the first step (as soon as some reasonable stack size is used) is very large. In order to benefit from the full search graph but at the same time to achieve a reasonable translation time for the second step, we need to prune the lattice.

We use OpenFST[6] for lattice manipulation and experiment with two types of lattice reduction: We "Prune" lattices at different pruning thresholds (p=0.1, 0.3 and 0.5) and we also try dropping all paths except N best ones using the OpenFST operation "ShortestPath" (N=1, 5, and 10).

The pruning threshold of 0.1 strikes harsh and the resulting lattice is rather close to the lattice with 1-best path. Bigger pruning thresholds leave more ambiguous options in the output lattices.

The results, forming the rows of Table 4, indicate that pruning gives best performance with smaller (i.e. harsher) thresholds whereas keeping n-best path performs better with a slightly higher number of paths (better BLEUs are obtained with N=5). This suggests that a good performance is obtained if the lattices contain just a few possible paths and not dozens or more of them.

Overall, passing lattices between the first and the second step does not help and none of our configurations matches the non-lattice baseline, 13.52.

5.2. Input Arc Weight and Optimization with Lattices

Using lattices in Moses input introduces one more weight to the model, the input arc weight (`-weight-i`) that reflects the importance of arc probabilities as given in the input lattice compared to other components of the translation system.

We perform our experiments in two variants that differ in the MERT tuning of the second step. See the first four columns of Table 4. In the first case, the second step is tuned on simple string input so the `-weight-i` is not determined and we manually set it to three different values, columns labelled "aw=0.2, 0.5 and 0.8". In the second and computationally more demanding case, MERT for the second step is run on lattices and the input arc weight is optimized together with all other model weights (column 4 in the table).

When producing lattices for the test set, we use a common setting in the 1^{st} step: both `-stack-limit` and `-cube-pruning-pop-limit` are set to default (1000) and the lattices are pruned as indicated in the row of the table. When producing lattices for the dev set (i.e. when tuning on lattices), `-stack-limit` and `-cube-pruning-pop-limit` are set to 100 and the lattices are not pruned at all. The similar parameter setting is used for "No Pruning" results.

There is an important difference between the two variants of tuning. Tuning on strings relies just on the man-made target side of our dev set; the input for the tuning is produced deterministically by converting fully inflected target forms ("back") to the LOF and MOT. The dev set then remains fully monotone and input and output tokens match 1-1. On the other hand, tuning on lattices requires to tune the second step on *machine translated* input as produced by the first step. The token-for-token correspondence is no longer guaranteed. The second step, restricted to monotone

[6] `http://www.openfst.org/twiki/bin/view/FST/WebHome`

	Arc Weight for Non-Lattice Optimization			Lattice Optimization	
Pruning Types	aw=0.2	aw=0.5	aw=0.8	without TAG LM	with TAG LM
No Pruning	10.05±0.40	10.40±0.43	10.72±0.42	12.98±0.54	13.11±0.48
p=0.1	13.11±0.50	13.12±0.51	13.13±0.51	13.12±0.52	13.49±0.50
p=0.3	12.85±0.52	12.85±0.52	12.84±0.52	12.92±0.53	13.40±0.50
p=0.5	12.25±0.48	12.31±0.48	12.34±0.49	12.77±0.51	13.31±0.48
N=1	12.98±0.54	12.98±0.54	12.98±0.54	12.98±0.55	13.30±0.52
N=5	13.10±0.53	13.12±0.53	13.12±0.53	13.07±0.55	13.40±0.52
N=10	13.06±0.52	13.06±0.51	13.07±0.52	13.08±0.53	13.43±0.51
Two-Step Baseline	13.52±0.51				13.75±0.51

Table 4: Results with unpruned and pruned lattice input on 2^{nd} step.

translation, then perhaps struggles to produce the reference instead of learning how to inflect best.

The results indicate that for the first variant, both pruning and n-best path experiments perform negligibly better with bigger arc weights.

The translation of lattices without any pruning, "No Pruning" results, indicate that the scores on lattice arcs are indeed important: if the lattice weight is trained properly or if the lattices are pruned to contain just a very few high scoring paths, the translation is better.

Overall, the experiment reveals that the sub-optimal tuning on strings (and manually setting the input arc weight) performs just as good as the proper tuning on lattices for translating pruned lattices.

5.3. Additional Tag LM

Our final experiment uses an additional language model over morphological tags in the 2^{nd} step. The second step translates from LOF+MOT to form and tag, benefiting from the language model of each factor. The BLEU score of the two-step baseline slightly increases from 13.52 to 13.75 when using this tag language model. The threshold-based pruning of lattices shows almost 0.4 gain in BLEU, rising from 13.12 up to 13.49. The lattice input with N-best paths shows a gradual increase in BLEU score with the increasing number of N-best paths.

6. Conclusion

We present our initial experiments with altering the two-step approach to machine translation in three directions. First, we re-examine, what is the best level of detail for the information that is passed between the first step (aimed at reordering and making lexical choices) and the second step (aimed at only inflecting the sentence). Second, we replace the phrase-based MT system in the first step with the hierarchical phrase-based system to cater for more complicated reordering patterns. And finally, we replace the simple string representation in the intermediate language with a lattice, so that many reordering options are passed between the first and the second step.

We see a little improvement in BLEU scores from using the hierarchical model. The results of lattice experiments could not exceed the two-step baseline in any of the experimental configurations. Possible improvements can come from some cleverer pruning of the lattices or in preparing the tuning data for the second step in a way that allows to provide lattices and at the same time ensures that the reference is very similar to one of the paths in the lattice.

7. Acknowledgments

This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013) and it is supported by the MosesCore project sponsored by the European Commissions Seventh Framework Programme (Grant Number 288487).

8. References

Bojar, Ondřej and Kos, Kamil. (2010). 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden, July. Association for Computational Linguistics.

Bojar, Ondřej, Mareček, David, Novák, Václav, Popel, Martin, Ptáček, Jan, Rouš, Jan, and Žabokrtský, Zdeněk. (2009). English-Czech MT in 2008. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March. Association for Computational Linguistics.

Bojar, Ondřej, Jawaid, Bushra, and Kamran, Amir. (2012). Probes in a taxonomy of factored phrase-based models. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 253–260, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, David. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, David. (2007). Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June.

Costa-jussà, Marta R., Crego, Josep M., Lambert, Patrik, Khalilov, Maxim, Fonollosa, José A. R., Mariño, José B., and Banchs, Rafael E. (2007). Ngram-Based Statistical Machine Translation Enhanced with Multiple Weighted Reordering Hypotheses. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 167–170, Prague, Czech Republic. Association for Computational Linguistics.

Deneefe, Steve, Hermjakob, Ulf, and Knight, Kevin. (2008). Overcoming vocabulary sparsity in mt using lattices. In *In Proceedings of AMTA*, pages 89–96. Association for Machine Translation in the Americas.

Dyer, Christopher, Muresan, Smaranda, and Resnik, Philip. (2008). Generalizing word lattice translation. In *Proceedings of the ACL*, pages 1012–1020, Columbus, Ohio, USA, June.

Koehn, Philipp and Hoang, Hieu. (2007). Factored translation models. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondřej, Constantin, Alexandra, and Herbst, Evan. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Minkov, Einat and Toutanova, Kristina. (2007). Generating complex morphology for machine translation. In *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL07*, pages 128–135.

Och, Franz Josef and Ney, Hermann. (2000). A Comparison of Alignment Models for Statistical Machine Translation. In *Proceedings of the 17th conference on Computational linguistics*, pages 1086–1090. Association for Computational Linguistics.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Popel, Martin and Žabokrtský, Zdeněk. (2010). TectoMT: Modular NLP Framework. In Loftsson, Hrafn, Rögnvaldsson, Eirikur, and Helgadottir, Sigrun, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304, Berlin / Heidelberg. Iceland Centre for Language Technology (ICLT), Springer.

Spoustová, Drahomíra, Hajič, Jan, Votrubec, Jan, Krbec, Pavel, and Květoň, Pavel. (2007). The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing, ACL 2007*, pages 67–74, Praha.

Stolcke, Andreas. (2002). Srilm - an extensible language modeling toolkit. In *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP) 2002*, pages 901–904.

Toutanova, Kristina, Suzuki, Hisami, and Ruopp, Achim. (2008). Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June. Association for Computational Linguistics.

Xu, Jia, Matusov, Evgeny, Zens, Richard, and Ney, Hermann. (2005). Integrated chinese word segmentation in statistical machine translation. In *In Proceedings of IWSLT*, Pennsylvania, USA.