Generalized Graphical Abstractions for Statistical Machine Translation

Karim Filali and **Jeff Bilmes**[*]
Departments of Computer Science & Engineering and Electrical Engineering
University of Washington
Seattle, WA 98195, USA
{karim@cs,bilmes@ee}.washington.edu

Abstract

We introduce a novel framework for the expression, rapid-prototyping, and evaluation of statistical machine-translation (MT) systems using graphical models. The framework extends dynamic Bayesian networks with multiple connected different-length streams, switching variable existence and dependence mechanisms, and constraint factors. We have implemented a new general-purpose MT training/decoding system in this framework, and have tested this on a variety of existing MT models (including the 4 IBM models), and some novel ones as well, all using Europarl as a test corpus. We describe the semantics of our representation, and present preliminary evaluations, showing that it is possible to prototype novel MT ideas in a short amount of time.

1 Introduction

We present a unified graphical model framework based on (Filali and Bilmes, 2006) for statistical machine translation. Graphical models utilize graphical descriptions of probabilistic processes, and are capable of quickly describing a wide variety of different sets of model assumptions. In our approach, either phrases or words can be used as the unit of translation, but as a first step, we have only implemented word-based models since our main goal is to show the viability of our graphical model representation and new software system.

There are several important advantages to a unified probabilistic framework for MT including: **(1)** the same codebase can be used for training and decoding without having to implement a separate decoder for each model; **(2)** new models can be prototyped quickly; **(3)** combining models (such as in a speech-MT system) is easier when they are encoded in the same framework; **(4)** sharing algorithms across different disciplines (e.g., the MT and the constraint-satisfaction community) is facilitated.

2 Graphical Model Framework

A *Graphical Model* (GM) represents a factorization of a family of joint probability distributions over a set of random variables using a graph. The graph specifies conditional independence relationships between the variables, and parameters of the model are associated with each variable or group thereof. There are many types of graphical models. For example, Bayesian networks use an acyclic directed graph and their parameters are conditional probabilities of each variable given its parents. Various forms of GM and their conditional independence properties are defined in (Lauritzen, 1996).

Our graphical representation, which we call *Multi-dynamic Bayesian Networks* (MDBNs) (Filali and Bilmes, 2006), is a generalization of dynamic Bayesian networks (DBNs) (Dean and Kanazawa, 1988). DBNs are an appropriate representation for sequential (for example, temporal) stochastic processes, but can be very difficult to apply when dependencies have arbitrary time-span and the existence of random variables is contingent on the val-

[*] This material was supported by NSF under Grant No. ISS-0326276.

Proceedings of NAACL HLT 2007, Companion Volume, pages 33–36,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

ues of certain variables in the network. In (Filali and Bilmes, 2006), we discuss inference and learning in MDBNs. Here we focus on representation and the evaluation of our new implementation and framework. Below, we summarize key features underlying our framework. In section 3, we explain how these features apply to a specific MT model.

- Multiple DBNs can be represented along with rules for how they are interconnected — the rule description lengths are fixed (they do not grow with the length of the DBNs).
- *Switching dependencies* (Geiger and Heckerman, 1996): a variable X is a *switching parent* of Y if X influences what type of dependencies Y has with respect to its other parents, e.g., an *alignment* variable in IBM Models 1 and 2 is switching.
- *Switching existence*: A variable X is "switching existence" with respect to variable Y if the value of X determines whether Y exists. An example is a *fertility* variable in IBM Models 3 and above.
- Constraints and aggregation: BN semantics can encode various types of constraints between groups of variables (Pearl, 1988). For example, in the construct $A \rightarrow B \leftarrow C$ where B is observed, B can constrain A and C to be unequal. We extend those semantics to support a more efficient evaluation of constraints under some variable order conditions.

3 GM Representation of IBM MT Models

In this section we present a GM representation for IBM model 3 (Brown et al., 1993) in fig. 1. Model 3 is intricate enough to showcase some of the features of our graphical representation but not as complex as, and thus is easier to describe, than model 4. Our choice of representing IBM models is not because we believe they are state of the art MT models—although they are still widely used in producing alignments and as features in log-linear models—but because they provide a good initial testbed for our architecture.

The topmost random variable (RV), ℓ, is a hidden switching existence variable corresponding to the length of the English string. The box abutting ℓ includes all the nodes whose existence depends on the value of ℓ. In the figure, $\ell = 3$, thus resulting in three English words $e_1, ..., e_3$, connected using a second-order Markov chain. To each English word e_i corresponds a conditionally dependent fertility ϕ_i, which indicates how many times e_i is used by words in the French string. Each ϕ_i in turn grants existence to a set of RVs under it. Given the fertilities (the figure depicts the case $\phi_1 = 3, \phi_2 = 1, \phi_3 = 0$), for each word e_i, ϕ_i French word RVs are granted existence and are denoted by the *tablet* $\tau_{i1}, \tau_{i2}, \ldots, \tau_{i\phi_i}$ of e_i. The values of τ variables need to match the actual observed French sequence $f_1, \ldots, f_m$. This is represented as a shared constraint between all the f, π, and τ variables which have incoming edges into the observed variable v. v's conditional probability table is such that it is one only when the associated constraint is satisfied. The variable $\pi_{i,k}$ is a switching dependency parent with respect to the constraint variable v and determines which f_j participates in an equality constraint with $\tau_{i,k}$.

In the null word sub-model, the constraint that successive permutation variables be ordered is implemented using the observed child w of π_{0i} and $\pi_{0(i+1)}$. The probability of w being unity is one only when the constraint is satisfied and zero otherwise.

The bottom variable m is a switching existence node (observed to be 6 in the figure) with corresponding French word sequence and alignment variables. The French sequence participates in the v constraint described above, while the alignment variables $a_j \in \{1, \ldots, \ell\}, j \in 1, \ldots, m$ constrain the fertilities to take their unique allowable values (for the given alignment). Alignments also restrict the domain of permutation variables, $\boldsymbol{\pi}$, using the constraint variable x. Finally, the domain size of each a_j has to lie in the interval $[0, \ell]$ and that is enforced by the variable u. The dashed edges connecting the alignment a variables represent an extension to implement an M3/M-HMM hybrid.[1]

4 Experiments

We have developed (in C++) a new entirely self-contained general-purpose MT training/decoding system based on our framework, of which we provide a preliminary evaluation in this section. Although the framework is perfectly capable of representing phrase-based models, we restrict ourselves to word-based models to show the viability of graphical models for MT and will consider different translation units in future work. We perform MT ex-

[1] We refer to the HMM MT model in (Vogel et al., 1996) as M-HMM to avoid any confusion.

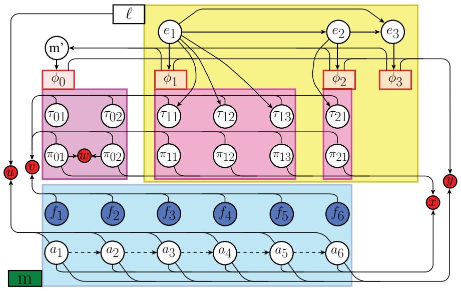

Figure 1: Unrolled Model 3 graphical model with fertility assignment $\phi_0 = 2, \phi_1 = 3, \phi_2 = 1, \phi_3 = 0$.

periments on a English-French subset of the Europarl corpus used for the ACL 2005 SMT evaluations (Koehn and Monz, 2005). We train an English language model on the whole training set using the SRILM toolkit (Stolcke, 2002) and train MT models mainly on a 10k sentence pair subset of the ACL training set. We test on the 2000 sentence test set used for the same evaluations. For comparison, we use the MT training program, GIZA++ (Och and Ney, 2003), the phrase-base decoder, Pharaoh (Koehn et al., 2003), and the word-based decoder, Rewrite (Germann, 2003).

For inference we use a backtracking depth-first search inference method with memoization that extends Value Elimination (Bacchus et al., 2003). The same inference engine is used for both training and decoding. As an admissible heuristic for decoding, we compute, for each node V with Conditional Probability Table c, the largest value of c over all possible configurations of V and its parents (Filali and Bilmes, 2006).

Decoder	**BLEU (%)**			
	500	**1000**	**1500**	**2000**
Rewrite	25.3	22.3	21.7	22.01
Pharaoh	20.4	18.1	17.7	18.05
M-HMM	19.9	16.9	15.6	12.5

Table 1: *BLEU scores on first 500, 1000, 1500, and 2000 sentences (ordered from shortest to longest) of the ACL05 English-French 2000 sentence test set using a 700k sent train set. The last row is our MDBN system's simulation of a M-HMM model.*

Table 1 compares MT performance between (1) Pharaoh (which uses beam search), (2) our system, and (3) Rewrite (hill-climbing). (1) and (2) make use of a fixed lexical table[2] learned using an M-HMM model specified using our tool, and neither uses minimum error rate training. (3) uses Model 4 parameters learned using GIZA++. This comparison is informative because Rewrite is a special purpose model 4 decoder and we would expect it to perform at least as well as decoders not written for a specific IBM model. Pharaoh is more general in that it only requires, as input, a lexical table from any given model.[3] Our MDBN system is not tailored for the translation task. Pharaoh was able to decode the 2000 sentences of the test set in 5000s on a 3.2GHz machine; Rewrite took 84000s, and we allotted 400000s for our engine (200s per sentence). We attribute the difference in speed and BLEU score between our system and Pharaoh to the fact Value Elimination searches in a depth-first fashion over the space of *partial configurations of RVs*, while Pharaoh expands *partial translation hypotheses* in a best-first search manner. Thus, Pharaoh can take advantage of knowledge about the MT problem's hypothesis space while the GM is agnostic with respect to the structure of the problem—something that is desirable from our perspective since generality is a main concern of ours. Moreover, the MDBN's heuristic and caching of previously explored subtrees have not yet proven able to defray the cost, associated with depth-first search, of exploring subtrees that do not contain any "good" configurations.

Table 2 shows BLEU scores of different MT models trained using our system. We decode using Pharaoh because the above speed difference in its favor allowed us to run more experiments and focus on the training aspect of different models. **M1, M2, M-HMM, M3, and M4** are the standard IBM models. **M2d** and **M-Hd** are variants in which the distortion between the French and English positions is used instead of the absolute alignment position. **M-Hdd** is a second-order **M-HMM** model (with distortion). **M3H** (see fig 1) is a variant of model 3 that uses first-order dependencies between alignment variables. **M-Hr** is another HMM model that uses the relative distortion between the current alignment and the previous one. This is similar to the model implemented by GIZA except we did

[2] Pharaoh's phrases are single words only.

[3] It does, however, use simple hard-coded distortion and fertility models.

	BLEU(%)			
	Giza train		MDBN train	
	10k	700k	10k	700k
M1	15.67	18.04	14.53	17.74
M2	15.84	18.52	15.74	
M2d	NA	NA	15.75	
M-HMM	NA	NA	15.87	
M-Hd	NA	NA	15.99	18.05
M-Hdd	NA	NA	15.55	
M-Hr	16.98	19.57	16.04	
M3	16.78	19.38	15.32	
M3H	NA	NA	15.67	
M4	16.81	19.51	15.00	
M4H	NA	NA	15.20	

Table 2: *BLEU scores for various models trained using GM and GIZA (when applicable). All models are decoded using Pharaoh.*

not include the English word class dependency. Finally, model **M4H** is a simplified model 4, in which only distortions within each tablet are modeled but a Markov dependency is also used between the alignment variables.

Table 2 also shows BLEU scores obtained by training equivalent IBM models using GIZA and the standard training regimen of initializing higher models with lower ones (we use the same schedules for our GM training, but only transfer lexical tables). The main observation is that GIZA-trained M-HMM, M3 and 4 have about 1% better BLEU scores than their corresponding MDBN versions. We attribute the difference in M3/4 scores to the fact we use a Viterbi-like training procedure (i.e., we consider a single configuration of the hidden variables in EM training) while GIZA uses pegging (Brown et al., 1993) to sum over a set of likely hidden variable configurations in EM.

While these preliminary results do not show improved MT performance, nor would we expect them to since they are on simulated IBM models, we find very promising the fact that this general-purpose graphical model-based system produces competitive MT results on a computationally challenging task.

5 Conclusion and Future Work

We have described a new probabilistic framework for doing statistical machine translation. We have focused so far on word-based translation. In future work, we intend to implement phrase-based MT models. We also plan to design better approximate inference strategies for training highly connected graphs such as IBM models 3 and 4, and some novel extensions. We are also working on new best-first search generalizations of our depth-first search inference to improve decoding time. As there has been increased interest in end-to-end task such as speech translation, dialog systems, and multilingual search, a new challenge is how best to combine the complex components of these systems into one framework. We believe that, in addition to the finite-state transducer approach, a graphical model framework such as ours would be well suited for this scientific and engineering endeavor.

References

F. Bacchus, S. Dalmao, and T. Pitassi. 2003. Value elimination: Bayesian inference via backtracking search. In *UAI-03*, pages 20–28, San Francisco, CA. Morgan Kaufmann.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

T. Dean and K. Kanazawa. 1988. Probabilistic temporal reasoning. *AAAI*, pages 524–528.

Karim Filali and Jeff Bilmes. 2006. Multi-dynamic Bayesian networks. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Dan Geiger and David Heckerman. 1996. Knowledge representation and inference in similarity networks and bayesian multinets. *Artif. Intell.*, 82(1-2):45–74.

Ulrich Germann. 2003. Greedy decoding for statistical Machine Translation in almost linear time. In *HLT-NAACL*, pages 72–79, Edmonton, Canada, May. ACL.

P. Koehn and C. Monz. 2005. Shared task: Statistical machine translation between European languages. pages 119–124, Ann Arbor, MI, June. ACL.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*, May.

S.L. Lauritzen. 1996. *Graphical Models*. Oxford Science Publications.

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

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

A. Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. Int. Conf. on Spoken Language Processing*.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*, pages 836–841.