A Discriminative Model for Semantics-to-String Translation

Aleš Tamchyna
Charles University in Prague
Malostranské náměstí 25
Prague, Czech Republic
tamchyna@ufal.mff.cuni.cz

Chris Quirk and **Michel Galley**
Microsoft Research
One Microsoft Way
Redmond, WA 98052
{chrisq,mgalley}@microsoft.com

Abstract

We present a feature-rich discriminative model for machine translation which uses an abstract semantic representation on the source side. We include our model as an additional feature in a phrase-based decoder and we show modest gains in BLEU score in an n-best re-ranking experiment.

1 Introduction

The goal of machine translation is to take source language utterances and convert them into fluent target language utterances with the same meaning. Most recent approaches learn transformations using statistical techniques on parallel data. Meaning equivalent representations of words and phrases are learned directly from natural data, as are other syntactic operations such as reordering. However, commonly used methods have a very simple view of the linguistic data. Each word is generally modeled independently, for instance, and the relations between words are generally captured only in fixed phrases or as syntactic relationships.

Recently there has been a resurgence of interest in unified semantic representations: deep analyses with heavy normalization of morphology, syntax, and even semantic representations. In particular, *Abstract Meaning Representation* (AMR, Banarescu et al. (2013)) is a novel representation of (sentential) semantics. Such representations could influence a number of natural language understanding and generation tasks, particularly machine translation.

Deeper models can be used for multiple aspects of the translation modeling problem. Building translation models that rely on a deeper representation of the input allows for a more parsimonious translation model: morphologically related words can be handled in a unified manner; semantically related concepts are immediately adjacent and available for modeling, etc. Language models using deep representations might help us model which interpretations are more plausible.

We present an initial discriminative method for modeling the likelihood of a target language surface string given source language deep semantics. This approach relies on an automatic parser for source language semantics. We use a system that parses into AMR-like structures (Vanderwende et al., 2015), and apply the resulting model as an additional feature in a translation system.

2 Related Work

There is a large body of related work on utilizing deep language representation in NLP and MT in particular. This is not surprising considering that such representations provide abstractions of many language-specific phenomena, effectively bringing different languages closer together.

A number of machine translation systems starting as early as the 1950s therefore used a form of transfer: the source sentences were parsed, and those parsed representations were translated into target representations. Finally text generation was applied. The level of analysis is somewhat arguable – sometimes it was purely syntactic, but in other cases it reached into the semantic domain.

One of the earliest architectures was described in 1957 (Yngve, 1957). More contemporary examples of such systems include KANT (Nyberg and Mitamura, 1992), which used a very deep representation close to an interlingua, early versions of SysTran and Microsoft Translator, or more recently TectoMT (Popel and Žabokrtský, 2010) for English→Czech translation.

AMR itself has recently been used for abstractive summarization (Liu et al., 2015). In this work, sentences in the document to be summarized are parsed to AMRs, then a decoding algorithm is run to produce a summary graph. The surface realization of this graph then constitutes the final sum-

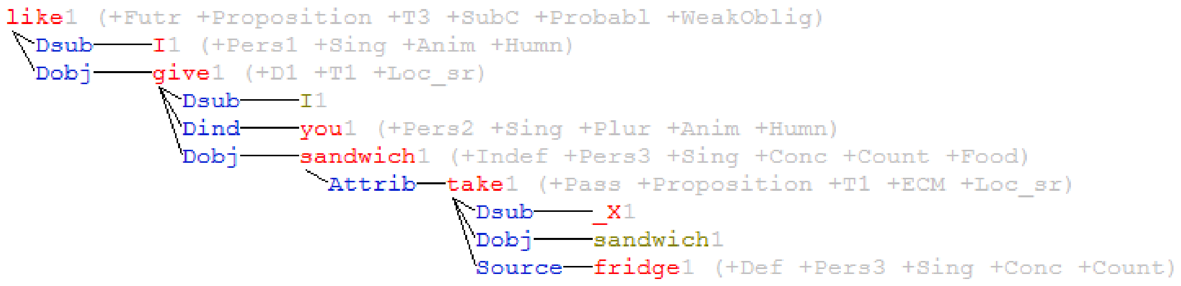
like1 (+Futr +Proposition +T3 +SubC +Probabl +WeakOblig)
Dsub
I1 (+Pers1 +Sing +Anim +Humn)
Dobj
give1 (+D1 +T1 +Loc_sr)
Dsub
I1
Dind
you1 (+Pers2 +Sing +Plur +Anim +Humn)
Dobj
sandwich1 (+Indef +Pers3 +Sing +Conc +Count +Food)
Attrib
take1 (+Pass +Proposition +T1 +ECM +Loc_sr)
Dsub
_X1
Dobj
sandwich1
Source
fridge1 (+Def +Pers3 +Sing +Conc +Count)

Figure 1: Logical Form (computed tree) for the sentence: *I would like to give you a sandwich taken from the fridge.*

mary.

(Jones et al., 2012) presents an MT approach that can exploit semantic graphs such as AMR, in a continuation of earlier work that abstracted translation away from strings (Yamada and Knight, 2001; Galley et al., 2004). While rule extraction algorithms such as (Galley et al., 2004) operate on trees and have also been applied to semantic parsing problems (Li et al., 2013), Jones et al. (2012) generalized these approaches by inducing synchronous hyperedge replacement grammars (HRG), which operate on graphs. In contrast to (Jones et al., 2012), our work does not have to deal with the complexities of HRG decoding, which runs in $O(n^3)$ (Jones et al., 2012), as our decoder is simply a phrase-based decoder.

Discriminative models have been used in statistical MT many times. Global lexicon model (Mauser et al., 2009) and phrase-sense disambiguation (Carpuat and Wu, 2007) are perhaps the best known methods. Similarly to Carpuat and Wu (2007), we use the classifier to rescore phrasal translations, however we do not train a separate classifier for each source phrase. Instead, we train a global model – similarly to Subotin (2011) or more recently Tamchyna et al. (2014). Features for our model are very different from previous work because they come from a deep representation and therefore should capture semantic relations between the languages, instead of surface or morpho-syntactic correspondences.

3 Semantic Representation

Our representation of sentence semantics is based on Logical Form (Vanderwende, 2015). LFs are labeled directed graphs whose nodes roughly correspond to content words in the sentence. Edge labels describe semantic relations between nodes. Additional linguistic information, such as verb subcategorization frames, definiteness, tense etc., is stored in graph nodes as *bits*.

Figure 1 shows a sentence parsed into the logical form. Nodes are represented by word lemmas. Relations include *Dsub* for deep subject, *Dobj* and *Dind* for direct and indirect objects etc. Bits are shown as flags in parentheses. Note that this graph may have cycles – for example, the *Dobj* of “take” is “sandwich”, but “take” is also the *Attrib* of “sandwich”. The verb “take” is also missing its obligatory subject which is replaced by the free variable *_X*.

The logical form can be converted using a sequence of rules to a representation which conforms to the AMR specification (Vanderwende et al., 2015). We do not use the full conversion pipeline in our work, so our semantic graphs are somewhere between the LF and AMR. Notably, we keep the *bits* which serve as important features for the discriminative modeling of translation.

4 Graph-to-String Translation

We develop models for semantic-graph-to-string translation. These models are essentially discriminative translation models, relying on a decomposition structure similar to both maximum entropy language models and IBM Models 1, 2 (Brown et al., 1993), and the HMM translation model (Vogel et al., 1996). In particular, we see translation as a process of selecting target words in order conditioned on source language representation as well as prior target words. Similar to the IBM Models, we see each target word as being generated based on source concepts, though in our case the concepts are semantic graph nodes rather than surface words. That is, we assume the existence of an alignment, though it aligns the target words to

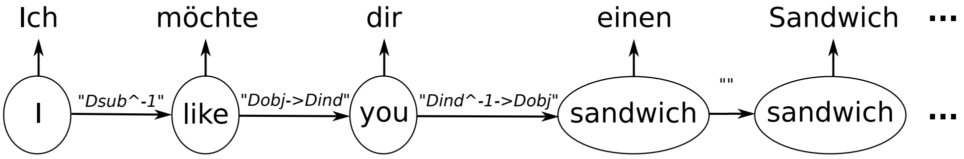

Figure 2: An example of the translation process illustrating several first steps of translating the sentence from Figure 1 into German ("*Ich möchte dir einen Sandwich...*"). Labels in italics correspond to the shortest undirected paths between the nodes.

source semantic graph nodes rather than surface words.

Our model views translation as generation of the target-side sentence given the source-side semantic graph. We assume a generative process which operates as follows. We begin in the virtual root node of the graph. At each step, we transition to a graph node and we generate a target-side word. We proceed left-to-right on the target side and we stop once the whole target sentence is generated. Figure 2 shows an example of this process.

Say we have a source semantic graph G with nodes $V = \{n_1..n_S\}$, edges $E \subset V \times V$, and a root node n_R for $R \in 1..S$. Then the likelihood of a target string $E = (e_1, ..., e_T)$ and alignment $A = (a_1, ..., a_T)$ with $a_i \in 0..S$ is as follows, with $a_0 = R$:

$$P(A, E|G) = \prod_{i=1}^{T} P(a_i|a_1^{i-1}, e_1^{i-1}, G) \quad P(e_i|a_1^{i}, e_1^{i-1}, G) \tag{1}$$

In this generative story, we first predict each alignment position and then predict each translated word. The transition distribution $P(a_i|\cdots)$ resembles that of the HMM alignment model, though the features are somewhat different. The translation distribution $P(e_i|\cdots)$ may take on several forms. For the purposes of alignment, we explore a simple categorical distribution as in the IBM models. For translation reranking, we instead use a feature-rich approach conditioned on a variety of source and target context.

4.1 Alignment of Semantic Graph Nodes

We have experimented with a number of techniques for aligning source-side semantic graph nodes to target-side surface words.

Gibbs sampling. We can attempt to directly align the target language words to the source language nodes using a generative HMM-style model. Unlike the HMM word alignment model (Vogel et al., 1996), the likelihood of jumping between nodes is based on the graph path between those nodes, rather than the linear distance.

Starting from the generative story of Equation 1, we make several simplifying assumptions. First we assume that the alignment distribution $P(a_i|\cdots)$ is modeled as a categorical distribution:

$$P(a_i|a_{i-1}, G) \propto \mathrm{c}(\textsc{label}(a_{i-1}, a_i))$$

The function LABEL*(u, v)* produces a string describing the labels along the shortest (undirected) path between the two nodes.

Next, we assume that the translation distribution is modeled as a set of categorical distributions, one for each source semantic node:

$$P(e_i|n_{a_i}) \propto \mathrm{c}(\textsc{lemma}(n_{a_i}) \to e_i)$$

This model is sensitive to the order in which source language information is presented in the target language.

The alignment variables a_i are not observed. We use Gibbs sampling rather than EM so that we can incorporate a sparse prior when estimating the parameters of the model and the assignments to these latent alignment variables. At each iteration, we shuffle the sentences in our training data. Then for each sentence, we visit all its tokens in a random order and re-align them. We sample the new alignment according to the Markov blanket, which has the following probability distribution:

$$P(t|n_i) \propto \frac{\mathrm{c}(\textsc{lemma}(n_i) \to t) + \alpha}{\mathrm{c}(\textsc{lemma}(n_i)) + \alpha L} \times \frac{\mathrm{c}(\textsc{label}(n_i, n_{i-1})) + \beta}{T + \beta P} \times \frac{\mathrm{c}(\textsc{label}(n_{i+1}, n_i)) + \beta}{T + \beta P} \tag{2}$$

L, P stand for the number of lemma/path *types*, respectively. T is the total number of tokens in the corpus. Overall, the formula describes the probability of the edge coming into the node n_i, the token emission and finally the outgoing edge. We evaluate this probability for each node n_i in the graph and re-align the token according to the random sample from this distribution.

α and β are hyper-parameters specifying the concentration parameters of symmetric Dirichlet priors over the transition and emission distributions. Specifying values less than 1 for these hyper-parameters pushes the model toward sparse solutions. They are tuned by a grid search which evaluates model perplexity on a held-out set.

Direct GIZA++. GIZA++ (Och and Ney, 2000) is a commonly used toolkit for word alignment which implements the IBM models. In this setting, we linearized the semantic graph nodes using a simple heuristic based on the surface word order and aligned them directly to the target-side sentences. We experimented with different symmetrizations and found that grow-diag-final-and gives the best results.

Composed alignments. We divided the alignment problem into two stages: aligning semantic graph nodes to source-side words and aligning the source- and target-side words (i.e., standard MT word alignment). We then simply compose the two alignments. For the alignment between source graph nodes and source surface words, we have two options: we can either train a GIZA++ model or we can use gold alignments provided by the semantic parser. For the second stage, we need to train a GIZA++ model.

We evaluated the different strategies by manually inspecting the resulting alignments. We found that the composition of two separate alignment steps produces clearly superior results, even if it seems arguable whether such division simplifies the task. Therefore, for the remaining experiments, we used the composition of gold alignment and GIZA++, although two GIZA++ steps performed comparably well.

4.2 Model

For our discriminative model, the alignment is assumed to be given. At training time, it is the alignment produced by the parser composed with GIZA++ surface word alignment. At test time, we compose the alignment between graph nodes and source surface tokens (given by the parser) with the bilingual surface word alignment provided by the MT decoder.

Turning to the translation distribution, we use a maximum entropy model to learn the conditional probability:

$$P(e_i|n_{a_i}, n_{a_{i-1}}, G, e_{i-k+1}^{i-1}) = \frac{\exp\left(\mathbf{w}\cdot\mathbf{f}(e_i, n_{a_i}, n_{a_{i-1}}, G, e_{i-k+1}^{i-1})\right)}{Z} \quad (3)$$

where Z is defined as

$$\sum_{e' \in GEN(n_{a_i})} \exp(\mathbf{w}\cdot\mathbf{f}(e', n_{a_i}, n_{a_{i-1}}, G, e_{i-k+1}^{i-1}))$$

The *GEN(n)* function produces the possible translations of the deep lemma associated with node n. We collect all translations observed in the training data and keep the 30 most frequent ones for each lemma. Our model thus assigns zero probability to unseen translations.

Because of the size of our training data, we used online learning. We implemented a parallelized (multi-threaded) version of the standard stochastic gradient descent algorithm (SGD). Our learning rate was fixed – using line search, we found the optimal rate to be 0.05. Our batch size was set to one; different batch sizes made almost no difference in model performance. We used online L1 regularization (Tsuruoka et al., 2009) with weight 1. We implemented feature hashing to further improve performance and set the hash length to 22 bits. We shuffled our data and split it into five parts which were processed independently and their final weights were averaged.

4.3 Feature Set

Our semantic representation enables us to use a very rich set of features, including information commonly used by both translation models and language models. We extract a significant amount of information from the graph node n_{a_i} aligned to the generated word:

- lemma,
- part of speech,
- all bits.

We extract the same features from the previous graph node ($n_{a_{i-1}}$), from the parent node. (If there are multiple parents in the graph, we break ties in a consistent but heuristic manner, picking the leftmost parent node according to its position in the source sentence) We also gather all the bits of the parent and the parent relation. These features may capture agreement phenomena.

We also look at the shortest path in the semantic graph from the previous node to the current one and we extract features which describe it:

- path length,
- relations (edges) along the path.

We use the lemmas of all nodes in the semantic graph as bag-of-word features, as well as all the surface words in the source sentence. We also extract lemmas of nodes within a given distance from the current node (i.e. graph context), as well as the relation that led to these nodes. Together, these features ground the current node in its semantic context.

An additional set of features handle the fact that source nodes may generate multiple target words, and the distribution over subsequent words should be different. We have a feature indicating the number of words generated from the current node, both in isolation, conjoined with the lemma, and conjoined with the part of speech. We also have a feature for each word previously generated by this same node, again in isolation, in conjunction with the lemma, and in conjunction with the part of speech. This helps prevent the model from generating multiple copies of same target word given a source node.

On the target side, we use several previous tokens as features. These may act as discriminative language model features.

During MT decoding, our model therefore must maintain state, which could present a computational issue. The language model features present similar complexity as conventional MT state, and the features about prior words generated from the same node require greater memory. Were this cost to become prohibitive, a simpler form of the prior word features would likely suffice.

5 Experiments

We tested our model in an n-best re-ranking experiment.

We began by training a basic phrase-based MT system for French→English on 1 million parallel sentence pairs and produced 1000-best lists for three test sets provided for the Workshop on Statistical Machine Translation (Bojar et al., 2013) – WMT 2009, 2010 and 2013. This system had a set of 13 commonly used features: four channel model scores (forward and backward MLE and lexical weighting scores), a 5-gram language model, five lexicalized reordering model scores (corresponding to different ordering outcomes), linear distortion penalty, word count, and phrase count. The system was optimized using minimum error rate training (Och, 2003) on WMT 2009.

Dataset	Baseline	+Semantics
WMT 2009 = devset	17.44	17.55
WMT 2010	17.59	17.64
WMT 2013	17.41	17.55

Table 1: BLEU scores of n-best reranking in French→English translation.

For reranking, we gathered 1000-best lists for the development and test sets. We added six scores from our model to each translation in the n-best lists. We included the total log probability, the sum of unnormalized scores, and the rank of the given output. In addition, we had count features indicating the number of words that were not in the GEN set of the model, the number of NULLs (effectively deleted nodes), and a count of times a target word appeared in a stopword list. In the end, each translation had a total of 19 features: 13 from the original features and 6 from this approach.

Next, we ran one iteration of the MERT optimizer on these 1000-best lists for all of the features. Because this was a reranking experiment rather than decoding, we did not repeatedly gather n-best lists as in decoding. The resulting feature weights were used to rescore the test n-best lists and evaluated the using BLEU; Table 1 shows the results. We obtained a modest but consistent improvement. Once the model is used directly in the decoder, the gains should increase as it will be able to influence decoding.

6 Conclusion

We have presented an initial attempt at including semantic features in a statistical machine translation system. Our approach uses discriminative training and a broad set of features to capture morphological, syntactic, and semantic information in a single model. Although our gains are not particularly large yet, we believe that additional effort on feature engineering and decoder integration could lead to more substantial gains.

Our approach is gated by the accuracy and consistency of the semantic parser. We have used a broad coverage parser with accuracy competitive to the current state-of-the-art, but even the state-of-the-art is rather low. It would be interesting to explore more robust features spanning multiple analyses, or to combine the outputs of multiple parsers. Even syntax-based machine translation systems are dependent on accurate parsers (Quirk and Corston-Oliver, 2006); deeper analyses are likely to be more dependent on parse quality.

In a similar vein, it would be interesting to evaluate the impact of morphological, syntactic, and semantic features separately. A careful feature ablation and exploration would help identify promising areas for future research.

We have only scratched the surface of possible integrations. Even this model could be applied to MT systems in multiple ways. For instance, rather than applying from source to target, we might evaluate in a noisy channel sense. That is, we could predict the source language surface forms given the target language translations. Furthermore, this would allow incorporation of a target semantic language model. This latter approach is particularly attractive, as it would explicitly model the semantic plausibility of the target. Of course, this would require target language semantic analysis: either we would be forced to parse n-best outcomes from some baseline system, or integrate the construction of target language semantics into the MT system. We believe that including such models of semantic plausibility holds great promise in preventing “word salad” outputs from MT systems: sentences that simply cannot be interpreted by humans.

Acknowledgements

We would like to thank Lucy Vanderwende and Arul Menezes for their helpful suggestions and insights into the logical form and abstract representations of meaning in general.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Eighth Workshop on Statistical Machine Translation*, WMT, pages 1–44, Sofia, Bulgaria.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Marine Carpuat and Dekai Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. Prague, Czech Republic.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proc. of HLT-NAACL*, pages 273–280.

Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. 2012. Semantics-based machine translation with hyperedge replacement grammars. In *Proc. of COLING*, pages 1359–1376.

Peng Li, Yang Liu, and Maosong Sun. 2013. An extended ghkm algorithm for inducing lambda-scfg. In *Proc. of AAAI*.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of NAACL 2015*, Denver, Colorado, June. Association for Computational Linguistics.

Arne Mauser, Sasa Hasan, and Hermann Ney. 2009. Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. pages 210–218, Suntec, Singapore.

Eric H. Nyberg and Teruko Mitamura. 1992. The kant system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 3*, COLING ’92, pages 1069–1073, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proc. of ACL*, pages 440–447, Hong Kong. ACL.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP Framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *IceTAL 2010*, volume 6233 of *LNCS*, pages

293–304. Iceland Centre for Language Technology (ICLT), Springer.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 62–69, Sydney, Australia, July. Association for Computational Linguistics.

Michael Subotin. 2011. An exponential translation model for target language morphology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 230–238, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aleš Tamchyna, Fabienne Braune, Alexander Fraser, Marine Carpuat, Hal Daumé III, and Chris Quirk. 2014. Integrating a discriminative classifier into phrase-based and hierarchical decoding. In *The Prague Bulletin of Mathematical Linguistics*.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 477–485. Association for Computational Linguistics.

Lucy Vanderwende, Arul Menezes, and Chris Quirk. 2015. An AMR parser for English, French, German, Spanish and Japanese and new AMR-annotated corpus. In *Proceedings of the 2015 NAACL HLT Demonstration Session*, Denver, Colorado, June. Association for Computational Linguistics.

Lucy Vanderwende. 2015. Nlpwin – an introduction. Technical Report MSR-TR-2015-23, March.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL*, pages 523–530.

V. H. Yngve. 1957. A framework for syntactic translation. *Mechanical Translation*, 4(3):59–65.