Semantics-Based Machine Translation with Hyperedge Replacement Grammars

Bevan JONES[1*] *Jacob ANDREAS*[2,3*] *Daniel BAUER*[3*]
Karl Moritz HERMANN[4*] *Kevin KNIGHT*[5]
(1) University of Edinburgh and Macquarie University
(2) University of Cambridge (3) Columbia University
(4) University of Oxford (5) Information Sciences Institute

Abstract

We present an approach to semantics-based statistical machine translation that uses synchronous hyperedge replacement grammars to translate into and from graph-shaped intermediate meaning representations, to our knowledge the first work in NLP to make use of synchronous context free graph grammars. We present algorithms for each step of the semantics-based translation pipeline, including a novel graph-to-word alignment algorithm and two algorithms for synchronous grammar rule extraction. We investigate the influence of syntactic annotations on semantics-based translation by presenting two alternative rule extraction algorithms, one that requires only semantic annotations and another that additionally relies on syntactic annotations, and explore the effect of syntax and language bias in meaning representation structures by running experiments with two different meaning representations, one biased toward an English syntax-like structure and another that is language neutral. While preliminary work, these experiments show promise for semantically-informed machine translation.

Title and Abstract in German

Semantikbasierte Maschinelle Übersetzung mit Hyperkantenersetzungsgrammatiken

Wir beschreiben einen Ansatz zur semantikbasierten statistischen maschinellen Übersetzung, der synchrone Hyperkantenersetzungsgrammatiken benutzt um in und aus graphgeformten Zwischenrepräsentationen zu übersetzen. Unseres Wissens ist dies die erste Arbeit in der natürlichen Sprachverarbeitung die synchrone kontextfreie Graphgrammatiken verwendet. Wir beschreiben Algorithmen für jeden Schritt der semantikbasierten Übersetzungskette, inklusive einem neuen Graph-zu-Wort Alinierungsalgorithmus und automatische Regelextraktionsalgorithmen für synchrone Grammatiken. Wir untersuchen den Effekt der syntaktischen Annotation auf semantikbasierte Übersetzung, indem wir zwei verschiedene Regelextraktionsalgorithmen vorstellen, einen, der lediglich semantische Annotationen erfordert und einen, der zusätzlich syntaktische Informationen verwendet. Wir untersuchen ausserdem den Einfluss von semantischen Repräsentationen die auf bestimmte Syntax und Sprache ausgerichted sind indem wir mit zwei verschiedenen Repräsentationen experimentieren: mit einer englischausgerichteten syntaxartigen Struktur und mit einer sprachneutralen Struktur. Unsere Arbeit zeigt dass semantikbasierte maschinelle Übersetzung vielversprechend ist.

[*] The authors contributed equally to this work and are listed in randomized order.

1 Introduction

In this paper, we introduce a model for semantic machine translation using a graph-structured meaning representation. While it has been claimed since the inception of machine translation that a semantic model is necessary to achieve human-like translation (Weaver, 1955; Bar-Hillel, 1960), most recent work in MT has instead focused on phrase-based approaches. Statistical phrase-based systems rely on large volumes of parallel training data to learn translation probabilities across two languages; while, given sufficient data, phrase-based systems can cope with some of the ambiguity problems identified by early MT researchers, they are limited by the underlying assumption that surface phrases can be translated without reference to syntax or meaning. Such systems often struggle to generate correct translations that involve non-local phenomena such as argument reorderings across languages, deep embeddings, empty categories and anaphora.

With the increasing availability of syntactically-annotated data in many languages, it has become possible to more directly integrate syntax into data-driven approaches. Such syntax-based SMT systems can automatically extract larger rules, and learn syntactic reorderings for translation (Yamada and Knight, 2001; Venugopal and Zollmann, 2006; Galley et al., 2004; Chiang, 2007; Zollmann et al., 2008; DeNero et al., 2009; Shen et al., 2010; Genzel, 2010).

However, many problems remain unsolved. For illustration of a specific phenomenon difficult to capture without an intermediate meaning representation, consider the following translation example using a state-of-the-art German→English SMT system [1]:

Source	**System output**	**Reference**
Anna fehlt ihrem Kater	**Anna is missing her cat*	Anna's cat is missing her

SMT systems are frequently unable to preserve basic meaning structures (e.g. "who does what to whom") across languages when confronted with verbs that realize their arguments differently. A system using an intermediate meaning representation need not suffer from this problem. Instead of learning many bilingual translation rules over all possible realizations of this pattern, it can rely on monolingual realizations to preserve meaning in translation.

Due to the recent emergence of large, multilingual, semantically annotated resources such as OntoNotes (Hovy et al., 2006), we believe the time is ripe for data-driven, semantics-based machine translation. In this paper we present a pilot statistical, semantic machine translation system which treats MT as a two-step process of analysis into meaning in the source language, and decoding from meaning in the target language.

Our system assumes that meaning representations are directed acyclic graphs; beyond that, it is completely agnostic with respect to the details of the formalism, including the inventory of node and edge labels used. Figure 1 illustrates a pipeline via one possible graph as semantic pivot. The proposed framework is flexible enough to handle numerous existing meaning representations, including the programming language syntax of the GeoQuery corpus (Wong and Mooney, 2006) (used for the experiments in this paper), the PropBank-style structures (Palmer et al., 2005) used for the CoNLL shared task on recognizing semantic dependencies (Hajič et al., 2009), and the Elementary Dependency Structures of the LOGON corpus (Oepen and Lønning, 2006).

[1] Google Translate, 08/31/2012

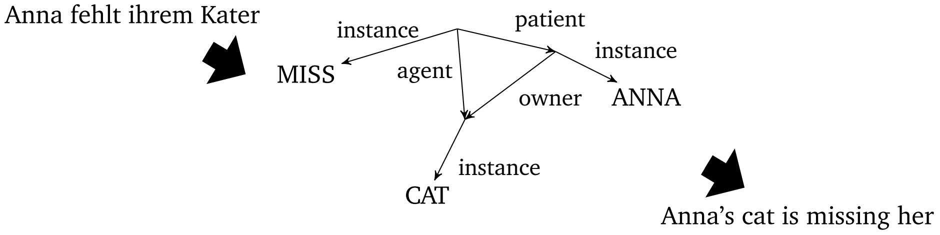


Figure 1: A string to meaning graph to string translation pipeline.

Experimental results demonstrate that our system is capable of learning semantic abstractions, and more specifically, to both analyse text into these abstractions and decode them back into text in multiple languages.

The need to manipulate graph structures adds an additional level of complexity to the standard MT task. While the problems of parsing and rule-extraction are well-studied for strings and trees, there has been considerably less work within the NLP community on the equivalent algorithms for graphs. In this paper, we use hyperedge replacement grammars (HRGs) (Drewes et al., 1997) for the basic machinery of graph manipulation; in particular, we use a synchronous HRG (SHRG) to relate graph and string derivations.

We provide the following contributions:

1. Introduction of string ⟺ graph transduction with HRGs to NLP
2. Efficient algorithms for
   - string–graph alignment
   - inference of graph grammars from aligned graph/string pairs
3. Empirical results from a working machine translation system, and analysis of that system's performance on the subproblems of semantic parsing and generation.

We proceed as follows: Section 2 explains the SHRG formalism and shows how it is used to derive graph-structured meaning representations. Section 3 introduces two algorithms for learning SHRG rules automatically from semantically-annotated corpora. Section 4 describes the details of our machine translation system, and explains how a SHRG is used to transform a natural language sentence into a meaning representation and vice-versa. Section 6 discusses related work and Section 7 summarizes the main results of the paper.

2 Synchronous Hyperedge Replacement Grammars

Hyperedge replacement grammars (Drewes et al., 1997) are an intuitive generalization of context free grammars (CFGs) from strings to hypergraphs. Where in CFGs strings are built up by successive rewriting of nonterminal *tokens*, in hyperedge replacement grammars (HRGs), nonterminals are *hyperedges*, and rewriting steps replace these nonterminal hyperedges with subgraphs rather than strings.

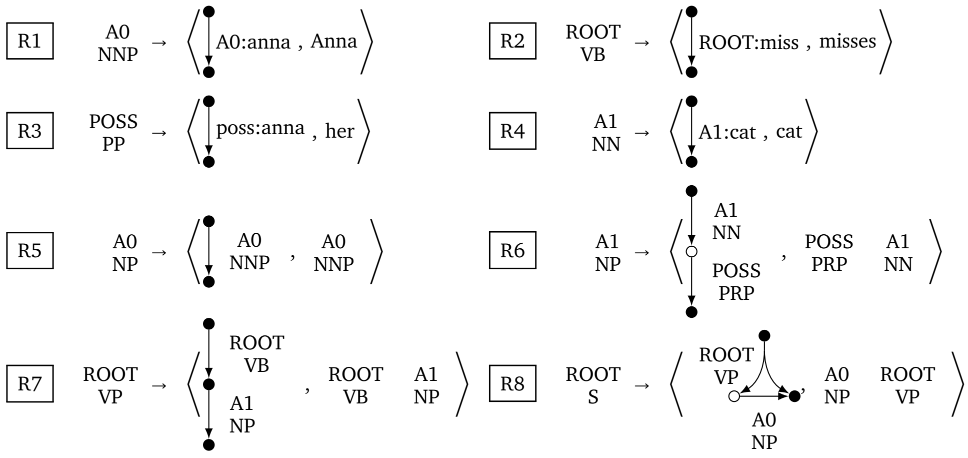
A hypergraph is a generalization of an graph in which edges may link an arbitrary number of nodes. Formally, a hypergraph over a set of edge labels C is a tuple $H = \langle V, E, l, X \rangle$, where V is a finite set of nodes, E is a finite set of edges, where each edge is a subset of V, $l : E \rightarrow C$ is a labeling function. $|e| \in \mathbb{N}$ denotes the *type* of a hyperedge $e \in E$ (the number of nodes connected by the edge). For the directed hypergraphs we are concerned with, each edge contains a distinguished source node and one or more target nodes.

A HRG over a set of labels C is a rewriting system $G = \langle N, T, P, S \rangle$, where N and $T \subset C$ are the finite sets of nonterminal and terminal labels ($T \cap N = \emptyset$), and $S \in N$ is the start symbol. P is a finite set of productions of the form $A \rightarrow R$, where $A \in N$ and R is a hypergraph over C, with a set of distinguished *external nodes*, X_R.

To describe the rewriting mechanism, let $H[e/R]$ be the hypergraph obtained by replacing the edge $e = (v_1 \cdots v_n)$ with the hypergraph R. The external nodes of R "fuse" to the nodes of e, $(v_1 \cdots v_n)$, so that R connects to $H[e/R]$ at the same nodes that e does to H. Note that $H[e/R]$ is undefined if $|e| \neq |X_R|$. Given some hypergraph H with an edge e, if there is a production $p : l_H(e) \rightarrow R \in G_P$ and $|X_R| = |e|$, we write $H \Rightarrow_p H[e/R]$ to indicate that p can derive $H[e/R]$ from H in a single step. We write $H \Rightarrow^*_G R$ to mean that R is derivable from H by G in some finite number of rewriting steps. The grammars we use in this paper do not contain *terminal* hyperedges, thus the yield of each complete derivation is a graph (but note that intermediate steps in the derivation may contain hyperedges).

A *Synchronous Hyperedge Replacement Grammar* (SHRG) is a HRG whose productions have pairs of right hand sides. Productions have the form $(A \rightarrow \langle R, Q \rangle, \sim)$, where $A \in N$ and R and Q are hypergraphs over $N \cup T$. $\sim$ is a bijection linking nonterminal mentions in R and Q. We call the R side of a rule the source and the Q side the target. Isolating each side produces a *projection* HRG of the SHRG. In general the target representation can be any hypergraph, or even a string since string can be represented as monadic (non-branching) graphs. Because we are interested in translation between MRs and natural language we focus on graph-string SHRGs. The target projection of such a SHRG is a context free string grammar. To ensure that source and target projection allow the same derivations, we constrain the relation $\sim$ such that every linked pair of nonterminals has the same label in R and Q.

Figure 2 shows an example SHRG with start symbol $\substack{\text{ROOT}\\\text{S}}$. External nodes are shaded black.



R1
A0 NNP → A0:anna , Anna
R2
ROOT VB → ROOT:miss , misses
R3
POSS PP → poss:anna , her
R4
A1 NN → A1:cat , cat
R5
A0 NP → A0 NNP , A0 NNP
R6
A1 NP → A1 NN POSS PRP , POSS PRP A1 NN
R7
ROOT VP → ROOT VB A1 NP , ROOT VB A1 NP
R8
ROOT S → ROOT VP A0 NP , A0 NP ROOT VP

Figure 2: A graph-string SHRG automatically extracted from the meaning representation graph in figure 3a using the SynSem algorithm. Note the hyperedge in rule R8.

The graph language captures a type of meaning representation in which semantic predicates and concepts are connected to their semantic arguments by directed edges. The edges are labeled with PropBank-style semantic roles (A0, A1, poss). Nonterminal symbols in this SHRG are complex symbols consisting of a semantic and a syntactic part, notated with the former above the latter.

Since HRG derivations are context free, we can represent them as trees. As an example, Figure 3c shows a derivation tree using the grammar in Figure 2, Figure 3a shows the resulting graph and Figure 3b the corresponding string. Describing graphs as their SHRG derivation trees allows us to use a number of standard algorithms from the NLP literature.

Finally, an *Adaptive Synchronous Hyperedge Replacement Grammar (ASHRG)* is a SHRG $G = \langle N, T, P^*, S, V\rangle$, where V is a finite set of variables. ASHRG production templates are of the same form as SHRG productions, $(A \rightarrow \langle R, Q\rangle, \sim)$, but $A \in N \cup V$ and $Q, R \in N \cup T \cup V$. A production template $p^* \in P^*$ is realised as a set of rules P by substituting all variables v for any symbol $s \in N \cup T$: $P = \{\forall_{v \in V} \forall_{s \in N \cup T} p^*[v/s]\}$. ASHRGs are a useful formalism for defining canonical grammars over the structure of graphs, with production templates describing graph structure transformations without regard to edge labels. We make use of this formalism in the production template R^* in Figure 4a.

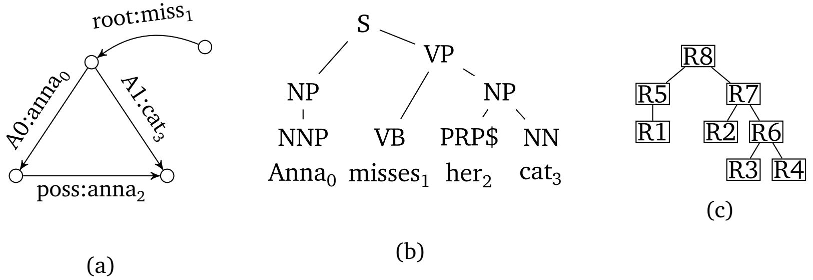


Figure 3: (a) an example meaning representation graph for the sentence ‘*Anna misses her cat.*’, (b) the corresponding syntax tree. Subscripts indicate which words align to which graph edges. (c) a SHRG derivation tree for (a) using the grammar Figure in 2.

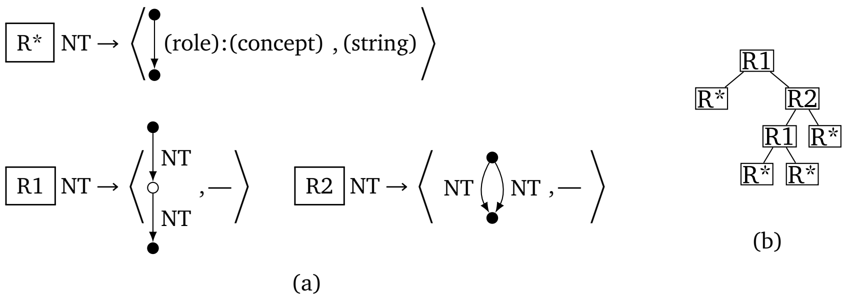


Figure 4: (a) The canonical grammar of width 2. R^* is a production template and values in parentheses denote variables as defined by the ASHRG formalism. (b) A SHRG derivation tree for the MR graph in Figure 3a using the canonical grammar in a, as created by the CanSem algorithm.

3 Learning Grammars from Annotated Data

3.1 Aligning Strings and Graphs

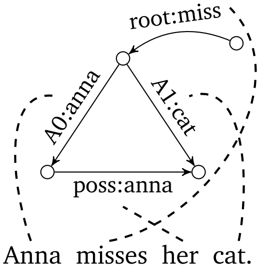


Figure 5: Edge-word alignment example.

Like much of SMT, alignments lie at the center of our semantics-based approach. However, in our case the alignments are between edges of the graph and words of the string. Figure 5 illustrates such an alignment. By listing out edge labels in a linear order, the graph-to-string alignment problem reduces to ordinary token-to-token alignment (Brown et al., 1990). We experiment with two strategies: (1) IBM Model 4 (M4) as implemented in GIZA++ (Och and Ney, 2003), and (2) a novel aligner that relies on the relative structure of the MR graph and the natural language syntax.

For M4, we traverse the graph in a fixed breadth first order to get a sequence of edge labels and feed this, along with the tokenized natural language string, to GIZA++. We then use the edge label order to map the aligned edge labels back to their respective edges.

We also experiment with a novel variant of IBM alignment Model 2 (Brown et al., 1990) that we call the dependency depth based aligner (DepDep, or DD for short) which uses depth within the graph and the dependency analysis of the natural language as location. Since the MR and the sentence describe the same thing, it seems reasonable to assume a certain degree of shared structure. To encode this notion, we place a Gaussian distribution over the difference between the depth of the graph edge and words in the dependency tree and weight the alignment choice by this probability. In this way, we favor aligning words to edges that are at a similar depth in the graph j to the depth of the word m in the dependency analysis.

The algorithm is concisely defined with the following equation, where a_i is the index of the edge aligned to the i^{th} word, f is a dependency parse with words f_i, e is a graph comprising edges e_i, and j_{a_i} and k_i are the edge and dependency depth of e_{a_i} and f_i, respectively.

$$p(f, a|e) = \prod_{i=1}^{n} p(j_{a_i}|k_i) p(f_i|e_{a_i}) \tag{1}$$

$$p(j_{a_i}|k_i) = \frac{\mathcal{N}(j_{a_i} - k_i|\mu, \sigma)}{\sum_{j'} \mathcal{N}(j' - k_i|\mu, \sigma)} \tag{2}$$

We estimate the mean μ and variance σ^2 of $p(j_{a_i}|k_i)$ with EM at the same time as the translation probabilities $p(f_i|e_{a_i})$.

3.2 Canonical Semantics Algorithm (CanSem)

Given word–edge alignments, we present two algorithms for rule extraction that both employ the same general strategy for rule extraction: They induce a single context-free derivation for each graph in the training data, and then extract rules from the aligned derivation trees and sentence spans.

Our first strategy for inducing a derivation of each training hypergraph (the "Canonical Semantics" Algorithm, or CanSem) is to specify, *a priori*, a minimal "canonical grammar" which is capable of producing every training example. A theorem by (Lautemann, 1988), proved by (Bodlaender, 1998), guarantees that a canonical grammar of width k is sufficient for graphs of maximum treewidth k. We extract a minimal grammar by incrementally increasing its width until the training data can be fully explained. The canonical grammar rules learned in this algorithm are effectively SHRG rule templates which ignore edge labels. Figure 4a shows a canonical grammar of width 2 needed to parse the graph in Figure 3a.

This grammar then allows us to immediately acquire a derivation tree given a graph alone (see Figure 4b); we can then use a standard technique (Galley et al., 2004) for acquiring a set of rules from an aligned derivation tree-string pair.

3.3 Syntactic Semantics Algorithm (SynSem)

Intuition suggests that additional linguistic information might aid in the selection of general, well-formed rules. Our second algorithm (the "Syntactic Semantics" Algorithm, or SynSem) is based on this assumption.

The procedure is described in Algorithm 1; to describe the notation, let each training example consist of (1) a sentence $S = s_1, s_2, \ldots, s_n$; (2) a constituency parse of S, defined by a set C of constituents; (3) a directed single-source connected hypergraph $H = (V, E)$; and (4) alignments $a : S \rightarrow E \cup \{ \qquad \}$. For convenience, denote the subspan of S contained in a constituent $c \in C$ (equivalently, the yield of c) as $S(c)$.

A constituent $c \in C$ is in the *frontier set* F if there exists some connected subgraph h of H such that $s \in S(c)$ if and only if $a(w) \in h$. Let $f(c)$ denote this h. c is in the *minimal frontier set* $\hat{F}$ if $c \in F$ and there exists no $c' \subset c \in F$.

Algorithm 1 EXTRACT-RULE

```
$R \leftarrow \emptyset$
**while** $\hat{F} \neq \emptyset$ **do**
    $c = pop(\hat{F})$
    $h$ is a new hyperedge with type matching $f(c)$
    $R = R \cup \{(h, f(c), c)\}$.
    $H = H[f(c)/h]$
    $S = S[S(c)/h]$
**end while**
```

On an abstract level, algorithm 1 matches minimal parse constituents to aligned graph components, and incrementally collapses these into nonterminals until the entire graph is consumed. Figure 2 shows the set of rules extracted by this algorithm from the MR graph, parse, and alignments in Figure 4.

In describing both algorithms, we have thus far assumed that every edge E is aligned to at least one word. As this is not always the case in practice, heuristics similar to those used in (Galley et al., 2006) may be used to attach the remaining edges.

4 Parsing and Translating with SHRG

4.1 Translation Pipeline

To build a translation system between two languages, we first extract an SHRG for each language from semantically annotated monolingual data using one of the algorithms from section 3. We then assign weights to each rule in the SHRG to transform it into a *probabilistic* SHRG, using one of the methods described below in Section 4.2.

Given a pair of weighted SHRGs, translation is a two-step process. First, we transform the source language string into an MR graph using the source SHRG (sometimes called "semantic parsing" or "analysis"). This is accomplished by parsing the string with the string projection of the SHRG and then applying the resultant derivation to generate the corresponding graph. The algorithm amounts to standard CKY string parsing with complexity $\mathcal{O}(n^3)$ in the size of the input.

We then transform the 1-best graph into the output string using the target SHRG (the "generation" task). This involves parsing the graph using the graph projection of the SHRG and then constructing the corresponding string yielded by the derivation. While parsing arbitrary graphs with SHRGs is NP-complete, we use a polynomial time chart parsing algorithms (which are exponential in the maximum size of the graph fragments on the rule right hand side) for connected graphs (Drewes et al., 1997).

In the case of the CanSem algorithm, we use a parser specialized for the canonical HRG which can be parsed even more efficiently in $\mathcal{O}(n^c)$, with c the maximum number of rule right hand side nodes (worst case $c = 3$ for experiments in this paper).

Finally, we rerank the natural language output by incorporating a language model (Heafield, 2011; Dyer et al., 2010). For the SynSem algorithm, this is integrated into the parsing algorithm via cube pruning (Chiang, 2007). In the CanSem algorithm reranking is performed on an k-best list of generated natural language output using a standard n-gram language model. Hypothesis meaning representations might be similarly reranked using a 'language model' defined on MR graphs, but we leave this for future work. In our experiments, language model weights were selected empirically based on initial evaluations of the development set.

4.2 Parameter Estimation

As with CFGs, there are two strategies for estimating the parameters of a probabilistic SHRG. One is to treat the derivations induced by the CanSem and SynSem algorithms as observed, and obtain maximum-likelihood estimates for the grammar by simply counting the number of times each production occurs in the training data.

The alternative approach is to employ the EM algorithm given a synchronous parse chart (Oates et al., 2003). Synchronous parsing extends graph parsing by identifying all possible derivations which yield both a specified string and a specified graph; the complexity of synchronous parsing is therefore approximately the product of string and graph parsing.

We initially tested both methods with both rule extraction algorithms on the development set.

For the CanSem algorithm, EM with a Dirichlet prior of 0.1 performed better, whereas the SynSem algorithm obtained better results using counting.

5 Evaluation

5.1 Data

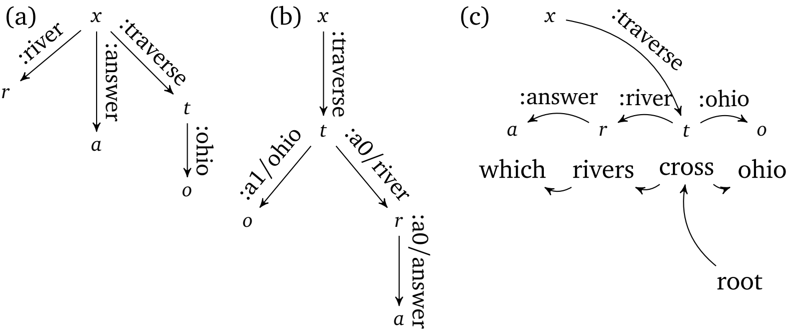


Figure 6: Two different ways of converting the GeoQuery Prolog expression to a MR graph: (a) language-neutral, (b) English-biased and (c) an illustration of how (b) matches the English dependency analysis.

Our experiments use the GeoQuery data set (Tang and Mooney, 2001), originally a parallel corpus of 880 English questions about US geography paired with Prolog style database queries and later translated into Chinese (Lu and Ng, 2011). For English there are gold Penn Treebank-style syntax annotations as well as gold alignments pairing every word with the best predicate in the query. For Chinese, we make use of automatic parses provided by the Stanford Parser (Levy and Manning, 2003).

The database queries—expressions in an unambiguous formal language—serve as a rough encoding of sentence meaning, which we use as our meaning representation in the machine translation pipeline. Though they do not, strictly speaking, encode a linguistic notion of semantics, a statistical MT system can still learn meaningful associations with this language-independent representation. (For instance the German *'Gib mir die Bevölkerung von Kalifornien!'* [*'Give me the population of California!'*] would match the the same Prolog query as English *'How many people live in California?'*.) For input to our system, we automatically translate the Prolog expressions into graphs.

We are interested in how differences between the syntax and semantic representation might impact the translation process, and we use two different graph representations to test this. One, shown in Figure 6a (corresponding to the question '*Which rivers cross Ohio?*'), is produced by only looking at the query expression itself (GQn). The second (GQe), shown in Figure 6b is a transformation of GQn to more closely match the English syntax using the gold alignments. Note that this reshaped graph better fits the assumptions of the SynSem algorithm and should, in theory, produce better SHRGs for English. It is less clear whether an English biased intermediate representation would perform better for translation, as it could conceivably hurt translation to and from other languages.

We use the standard 600 train/280 test sentence split (Kwiatkowski et al., 2010), and run 10

	Prec.	Rec.	f_1
DD	**74.8**	46.9	**57.7**
M4	54.6	**53.7**	54.1

Table 1: Evaluation of English alignment, vs. gold alignments

fold cross-validation on the training data during development. We also use the standard list of named entities paired with the corresponding edges to create some fallback rules for handling previously unseen named entities.

5.2 Alignment

Table 1 shows results for the alignment algorithms described in Section 3.1 on English and the GQN MR. We report precision, recall and f_1-measure on alignment pairs. The DEPDEP algorithm (DD) performs somewhat better than IBM Model 4 (M4) with respect to f_1, and substantially better in terms of precision.

5.3 Analysis: String to Graph

We use the Smatch measure (Cai and Knight, 2012) to evaluate analysis into MR graphs, which is essentially an f_1-score on edge labels under the optimal mapping of hypothesis nodes onto reference nodes. Table 2 shows results for the analysis task in both English and Chinese. We report results of 10-fold cross validation on the training set, reserving the test data for the evaluation of the end-to-end MT system. We apply both rule extraction algorithms to the GQN data set; for English, where we have gold alignments and gold parses (which allow us to obtain DD alignments), we also vary the alignment model. We observe that the SYNSEM procedure uniformly outperforms CANSEM for analysis. These results highlight the importance of alignment quality: Gold alignments unsurprisingly lead to improved analysis, and in accordance with our expectations the syntactically-guided DD alignments appear to help SYNSEM but not CANSEM.

	CANSEM			SYNSEM		
	M4	DD	GOLD	M4	DD	GOLD
EN	67.9	56.4	72.4	81.5	81.8	84.4
ZH	67.8	–	–	76.8	–	–

Table 2: Evaluation of analysis (f_1), vs. gold MRs in development set

	CANSEM			SYNSEM		
	M4	DD	GOLD	M4	DD	GOLD
EN	51.89	48.82	55.24	52.47	42.91	53.3
ZH	50.28	–	–	45.82	–	–

Table 3: Evaluation of generation (BLEU), vs. gold strings in development set

5.4 Generation: Graph to String

We evaluate text generated from gold MR graphs using the well-known BLEU measure (Papineni et al., 2002). Table 3 shows results for English (EN) and Chinese (ZH), varying rule extraction and alignment model as before. As before, M4 alignments help CanSem more than DD alignments; however, here the trend also carries through to SynSem. Also in contrast to the analysis results, the two systems perform comparably on their best English results, and CanSem outperforms SynSem on Chinese.

While not targeted directly at the generation task (and not comparable to the existing literature, which reports BLEU scores on the test set), these results are promising: They are close to state-of-the-art for generation on the GeoQuery data set, and future research might focus on optimizing generation specifically.

5.5 Translation: String to String

Finally, Table 4 shows results for the end-to-end machine translation system for English to Chinese, evaluated on the test set. We again experiment with both rule extraction algorithms in English (because DD alignments are not available for Chinese, Chinese rule extraction always uses M4).

Here CanSem substantially outperforms SynSem, regardless of the data set and the choice of alignment algorithm. Also notable is the fact that the switch from GQN to GQE hurts performance with CanSem but improves it with SynSem.

	CanSem		SynSem	
	M4	DD	M4	DD
GQN	42.74	36.84	28.22	28.34
GQE	38.88	35.14	32.24	31.20

Table 4: Evaluation of translation (BLEU), vs. gold strings in the test set

5.6 Discussion

Several broad trends are apparent from these experimental results. The first is a partial confirmation of our hypothesis that syntactic information (in various forms) is useful in guiding the acquisition and application of semantic grammars: this is apparent in SynSem's gains on analysis and possibly generation, and the fact that SynSem performs better on GQE.

In this light the fact that CanSem performs comparatively better on translation is somewhat surprising—we would expect overall translation results to be improved as a result of improved analysis and comparable generation. We hypothesize that the discrepancy in translation scores is due to the consistency of the grammars learned by CanSem: Because it induces a standard derivation for MRs regardless of the source language, any incorrect rules that it learns are nonetheless shared across languages.

We also observe that the system generates many output sentences that have identical meaning but markedly different syntax and lexical choice from the reference translation (Figure 8). An inspection of the *k*-best list (Figure 7) for one translation input reveals various such candidates. This confirms that the translation pipeline works as expected: Sentences in the source

Reference	what state has the sparsest population density
$k1$	what state has the least population density
$k2$	which state has the least population density
$k3$	what is the state with the smallest population
$k4$	what state has the smallest population
$k5$	what is the state with the smallest population density

Figure 7: k-best list for a sample CanSem translations

Sample sentence	Reference
what is the density of texas	what is the population density of texas
what is the tallest mountain in america	what is the highest mountain in the us
what rivers the most running through it	which state has the most rivers running through it

(a) CanSem

Sample sentence	Reference
give me cities with the largest population	what city has the largest population
what is the population of washington	how many people live in washington
what are in washington	how many rivers in washington

(b) SynSem

Figure 8: Sample translation output.

language are analyzed into a language-independent meaning representation, and that meaning representation is then used to generate a semantically equivalent sentence in the target language.

The scores reported for SynSem are especially heartening in light of the fact that a standard phrase-based SMT system (Koehn et al., 2007), trained and tuned on the same corpus, achieves a BLEU score of 45.13 for zh–en translation. Note that the semantic translation task (at least as formulated here) is strictly harder than direct translation, as the test set contains numerous sentences annotated with identical meaning representations but different natural language realizations. We are optimistic about the potential for an extended version of the current system in which generation is conditioned on both semantics and source language.

6 Related Work

We view our semantics-based approach to MT as a continuation of recent work in statistical MT (SMT) that abstracts away from the surface string level by capturing syntactic reorderings in translation (Yamada and Knight, 2001; Gildea, 2003; Eisner, 2003; Collins et al., 2005), or using larger syntactic fragments instead of phrases (Galley et al., 2004, 2006; Chiang, 2007). These systems combine the benefits of rule-based MT and SMT by defining their translation model using syntactic translation rules from the source syntax, into the target syntax, or both. Our syntax-driven approach to rule extraction is inspired by (Chiang, 2007, 2010), while the canonical grammar approach is based on (Galley et al., 2004, 2006). However, we induce synchronous graph grammars between surface form and meaning representation, instead of transfer rules between source and target form. As with other translation work using synchronous tree grammars, such as synchronous TSG (Chiang, 2010) and synchronous TAG

(DeNeefe and Knight, 2009), our SHRGs can also be applied in both directions.

However, none of these SMT approaches use an intermediate semantic representation. A lot of research has been done in the early days of MT on translation systems using such representations (Uchida, 1987; Nirenburg, 1989; Landsbergen, 1989). These systems usually required hand-crafted rules and large knowledge bases and do not learn translation models from data automatically. Until recently, because of their good performance especially in narrow domains, rule-based MT was the predominant paradigm in deployed MT systems. In contrast, while our system adopts a semantic transfer based paradigm, we learn weighted transfer rules into and from the meaning representation automatically to build a true statistical semantics-based MT system.

In the semantic parsing literature, there are other learning based approaches to analysis into meaning representations. Zettlemoyer and Collins (2005) use an automatically induced, semantically augmented CCG and a log-linear model to parse into lambda expressions, and Ge and Mooney (2005) integrate syntactic parsing with semantic parsing for recovering Prolog queries. Lu et al. (2008) learn a generative model over tree shaped meaning representation and natural language sentences. Wong and Mooney (2006)'s WASP system is similar to ours because it draws on techniques from SMT, using word alignment algorithms to learn synchronous CFGs which translate between syntax and semantics. In fact, Jones et al. (2012) recasts many semantic parsing approaches as tree transduction, which is closely related to synchronous grammar parsing (Shieber, 2004). To our knowledge we are the first to address semantic parsing into graph-based representations as a learning task using synchronous graph grammars.

In generation, learning the parameters of statistical generation models is popular, but not much attention has been paid to the scenario where no handwritten rules exist or the mapping between semantic structure and output language is unknown in the training data (the scenario we assume in this paper). The WASP system (Wong and Mooney, 2006) can also be used as a generator. Lu and Ng (2011) automatically learn to generate English and Chinese from sentences paired with lambda calculus. Other examples are (Varges and Mellish, 2001) who learn a semantic grammar form a semantically annotated treebank automatically, and (DeVault et al., 2008) who infer a TAG for generation automatically from semantically annotated example sentences.

Formal language approaches to probabilistic tree transformation are popular (e.g. in syntax-based MT) and recently a formulation of such methods as tree transducers (Comon et al., 2007) has gained prominence in NLP (Knight and May, 2009; Graehl and Knight, 2004). In contrast, little work has been done on methods for graphs in NLP. The standard model for graph-shaped meaning representations in NLP are feature structures, which can be constructed from strings using unification grammars (Moore, 1989). However, while powerful in representation, unification grammars have unfavorable algorithmic properties, lack an intuitive probabilistic extension, and require hand-built rules. Other formal devices to accept and transduce feature structure graphs have rarely been discussed. Notable exceptions are Quernheim and Knight (2012) who discuss formal devices to accept and transduce feature structure graphs, and Bohnet and Wanner (2010) who present a toolkit for manually engineering graph-to-string transducer rules for natural language generation. We believe that we are the first to use hyperedge replacement grammars in the NLP literature and can only refer to the formal HRG survey by (Drewes et al., 1997).

7 Conclusion

We have introduced a new model for semantically-driven statistical machine translation using graph-structured meaning representations. Our approach is based on the class of weighted synchronous hyperedge replacement grammars, a rewriting formalism for graph-string pairs that intuitively extends context-free grammars. We have described unsupervised algorithms for string-graph alignment, and two algorithms for automatic SHRG learning given these alignments.

We have evaluated a semantic machine translation system on the GeoQuery data set, using grammars acquired using each of these algorithms. The results of this evaluation provide a working demonstration of the effectiveness of our machine translation model, and a characterization of the extent to which syntactic information may be used to improve the effectiveness semantic MT.

We hope that our work will motivate further research on the applications of graph grammars to basic problems in natural language processing research. Immediate extensions of the research presented here include integration of a re-ranking model for hypothesized MRs (analogous to language modeling for strings), investigation of other corpora and meaning representation formalisms, and more sophisticated probabilistic models for scoring SHRG derivations. More broadly, our results suggest that SHRGs might be an effective tool for the individual problems of semantic parsing and generation, and indeed for any phenomenon in natural language which can be represented with directed graphs.

Acknowledgments

This research was supported by ARO grant W911NF-10-1-0533 and DARPA/IPTO Contract No. HR0011-12-C-0014.

References

Bar-Hillel, Y. (1960). The present status of automatic translation of languages. In Alt, F. L., editor, *Advances in Computers*. Academic Press, New York.

Bodlaender, H. L. (1998). A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45.

Bohnet, B. and Wanner, L. (2010). Open source graph transducer interpreter and grammar development environment. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, Valletta, Malta. European Language Resources Association (ELRA).

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Cai, S. and Knight, K. (2012). Smatch: an evaluation metric for semantic feature structures. submitted.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D. (2010). Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1443–1452.

Collins, M., Koehn, P., and Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 531–540.

Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., and Tommasi, M. (2007). *Tree Automata Techniques and Applications*. INRIA.

DeNeefe, S. and Knight, K. (2009). Synchronous tree adjoining machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 727–736. Association for Computational Linguistics.

DeNero, J., Pauls, A., and Klein, D. (2009). Asynchronous binarization for synchronous grammars. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 141–144.

DeVault, D., Traum, D., and Artstein, R. (2008). Practical grammar-based NLG from examples. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG '08, pages 77–85.

Drewes, F., Habel, A., and Kreowski, H. (1997). Hyperedge replacement graph grammars. In Rozenberg, G., editor, *Handbook of Graph Grammars and Computing by Graph Transformation*, pages 95–162. World Scientific.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL*.

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2*, pages 205–208. Association for Computational Linguistics.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *ACL 2006*.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule. In *Proceedings of HLT/NAACL*, volume 4, pages 273–280. Boston.

Ge, R. and Mooney, R. J. (2005). A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 9–16. Association for Computational Linguistics.

Genzel, D. (2010). Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 376–384. Association for Computational Linguistics.

Gildea, D. (2003). Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 80–87. Association for Computational Linguistics.

Graehl, J. and Knight, K. (2004). Training tree transducers. In *HLT-NAACL*, pages 105–112.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK. Association for Computational Linguistics.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Jones, B., Johnson, M., and Goldwater, S. (2012). Semantic parsing with bayesian tree transducers. In *ACL 2012*.

Knight, K. and May, J. (2009). Applications of weighted automata in natural language processing. In *Handbook of Weighted Automata*. Springer.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., and Steedman, M. (2010). Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233. Association for Computational Linguistics.

Landsbergen, J. (1989). The Rosetta project. In *Machine Translation Summit II*, Munich, Germany.

Lautemann, C. (1988). Decomposition trees: Structured graph representation and efficient algorithms. In *Proceedings of the 13th Colloquium on Trees in Algebra and Programming*, CAAP '88, pages 28–39, London, UK, UK. Springer-Verlag.

Levy, R. and Manning, C. D. (2003). Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of ACL 2003*, pages 439–446.

Lu, W., Ng, H., Lee, W., and Zettlemoyer, L. (2008). A generative model for parsing natural language to meaning representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 783–792. Association for Computational Linguistics.

Lu, W. and Ng, H. T. (2011). A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1611–1622. Association for Computational Linguistics.

Moore, R. C. (1989). Unification-based semantic interpretation. In *Proceedings of ACL 1989*, pages 33–41.

Nirenburg, S. (1989). New developments in knowledge-based machine translation. In Alatis, J. E., editor, *Georgetown University Round Table on Languages and Linguistics 1989: “Language teaching, testing, and technology: lessons from the past with a view toward the future”*, pages 344–357. Georgetown University Press.

Oates, T., Doshi, S., and Huang, F. (2003). Estimating maximum likelihood parameters for stochastic context-free graph grammars. In *Inductive Logic Programming: 13th International Conference*, pages 281–298.

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

Oepen, S. and Lønning, J. (2006). Discriminant-based MRS banking. In *5th International Conference on Language Resources and Evaluation*.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, pages 311–318.

Quernheim, D. and Knight, K. (2012). Towards Probabilistic Acceptors and Transducers for Feature Structures. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Shen, L., Xu, J., and Weischedel, R. (2010). String-to-dependency statistical machine translation. *Computational Linguistics*, 36(4):649–671.

Shieber, S. M. (2004). Synchronous grammars as tree transducers. In *Proceedings of TAG+7*, pages 88–95.

Tang, L. R. and Mooney, R. J. (2001). Using multiple clause constructors in inductive logic programming for semantic parsing. In *12th European Conference on Machine Learning*.

Uchida, H. (1987). ATLAS: Fujitsu machine translation system. In *Machine Translation Summit*, Japan.

Varges, S. and Mellish, C. (2001). Instance-based natural language generation. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Venugopal, A. and Zollmann, A. (2006). Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation. New York City: Association for Computational Linguistics*, pages 138–141.

Weaver, W. (1955). Translation. In *Machine translation of languages*, volume 14, pages 15–23. MIT Press, Cambridge, MA.

Wong, Y. W. and Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL ’06, pages 439–446.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530.

Zettlemoyer, L. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 658–666, Arlington, Virginia. AUAI Press.

Zollmann, A., Venugopal, A., Och, F., and Ponte, J. (2008). A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 1145–1152.