Toward Semantic Machine Translation

Jacob Andreas

Submitted in partial fulfillment of the
requirements for the degree
of Bachelor of Science
in the School of Engineering and Applied Science

COLUMBIA UNIVERSITY

2012

ABSTRACT

Toward Semantic Machine Translation

Jacob Andreas

This thesis presents a novel approach to interlingual machine translation using λ-calculus expressions as an intermediate representation. It investigates and extends existing algorithms which learn a combinatorial category grammar for semantic parsing, and introduces two new algorithms for generation out of logical forms inspired by that semantic parser. The results of a set of new experiments for generation and parsing are described, as well as an evaluation of the performance of a semantic translation system created by joining the semantic parser and generator together. Experimental results demonstrate that under certain conditions, this semantic model achieves better performance than a standard phrase-based statistical machine translation system in both an automated evaluation of translation output and a manual evaluation of adequacy and fluency.

Table of Contents

List of Tables

A thesis, even one of modest size, is almost inevitably a group undertaking; mine is no exception. Foremost thanks are due to my adviser Michael Collins for his expert guidance on this project. I am also deeply grateful to Kathleen McKeown, for giving me a chance to dirty my hands with NLP research long before I was qualified to do so, and to Nizar Habash and Owen Rambow, for introducing me to machine translation. This project would have been impossible without advice and code from Alexander Rush, Tom Kwiatkowski, Luke Zettlemoyer, Hwee Tou Ng and Wei Lu, and without direction from Laura Furst and Danielle Wong-Asuncion through the Cretan labyrinth of Columbia bureaucracy. I am similarly indebted my manual evaluators (and sometime roommates) John Croll, Melissa Bermudez, Conor Cashel and Dylan MacGowan; my major advisers Steve Bellovin and Al Aho; and finally friends and family too innumerable to mention, to whose intellect and companionship I owe much more than this thesis.

Translations of the *Zhuangzi*, with the exception of the first, are from [dBB00].

To C.J.A.

荃者所以在魚得魚而忘荃
蹄者所以在兔得兔而忘蹄
言者所以在意得意而忘言
吾安得夫忘言之人而與之言哉

庄子

The trap is for fish; having the fish you can forget the trap.
The snare is for rabbits; having the rabbit you can forget the snare.
Language is for meaning; having meaning you can forget language.
Where can I find a man who has forgotten language, so I can talk to him?

Zhuangzi

Chapter 1

Introduction

> I have a big tree of the kind men call *shu*. Its trunk is too gnarled and bumpy [...] its branches too bent and twisty [...] You could stand it by the road and no carpenter would look at it twice. Your words, too, are big and useless, and so everyone alike spurns them!
>
> ZHUANGZI

A semantically-informed model of machine translation can improve both the adequacy and fluency of translation outputs. In this thesis, I present a novel approach to interlingual machine translation using λ-calculus expressions as an intermediate meaning representation. In doing so, I hope to point toward a means of unifying now-defunct interlingual models of machine translation and the data-driven methods that replaced them.

Certainly, the time has come to fundamentally reconsider our approach to automated translation: The current phrase-based, statistical method, though it has proven superior to the hand-tuning that preceded it as the dominant paradigm [CBKMZ11], is satisfying neither as a model of the process underlying human translation, nor—more importantly—for practical purposes, as a mechanism for transforming sentences in one language into meaningful, well-formed sentences in another.[1] Though scores against standard evaluations

[1] As recently as 2010, the best-performing system at the Workshop on Machine Translation was judged to produce "acceptable" output for French–English translation only 54% of the time. [CBKM+10]

improve slowly from year to year, it's unclear if these scores, whether assigned by humans or automated metrics, correspond to genuine improvements in translation quality [TSM03, CBOK06].

Recent years have seen promising success in syntax-sensitive machine translation, either by building it into the phrase-based model [XM04] or by learning transfer rules as an alternative to phrase tables [Chi05]. Thus it's reasonable to ask whether an even higher level of abstraction might provide even further gains in both the clarity and correctness of automatic translation. Constant recourse to memorized rules of syntax is the mark of a skilled language learner, but not a fluent speaker; the native instead interprets and translates in terms of the *meaning* of the sentence observed. If we ever wish to have translation systems which are truly fluent, it may be that they, too, must learn to model meaning.

This is by no means a new idea; indeed, some of the very earliest systems for machine translation were driven by semantic models. But those systems, like much of the other rule-based artificial intelligence of that generation, were brittle and limited, and were quickly outpaced when statistical techniques were developed.

I think we are ready to try semantics again, and to unify the old-fashioned interlingual approach with modern statistical techniques. The system I present in this thesis is not a general-purpose translation system: It is itself brittle and narrow, and it does not address a number of crucial issues (such as the representation of meaning and the interpretation of context) which must be solved in order for such a system to be produced. My goal is simply to demonstrate that statistically driven, semantics-based parsing is already possible.

The study of algorithms for automatically translating text from one human language to another has been recognized as an an important application of computers from the very earliest days of computer science. Machine translation was envisioned by none less than Alan Turing and Warren Weaver and identified as a crucial part of American intelligence strategy during the Cold War [Hut97] and the modern "War on Terror", and is used (occasionally to great hilarity) by individuals and businesses throughout the world today. In recent years translation has become central to numerous humanitarian efforts surrounding education and emergency response [HBG+11]. Even science fiction, rather than assuming some universal

language, tends to rely on a ubiquitous translation service.[2]

It seems like translation ought to be a straightforward task: Natural language, after all, is generated according to rules so simple small children learn them effortlessly, and "all" a machine translation system needs in order to function is to be told (or discover) those rules for itself. Why, then, should we believe that a model of semantics is necessary for successful translation?

In 1949, Warren Weaver wrote:

> Thus it may be true that the way to translate from Chinese to Arabic, or from Russian to Portuguese, is not to attempt the direct route, shouting from tower to tower. Perhaps the way is to descend, from each language, down to the common base of human communication—the real but as yet undiscovered universal language—and then re-emerge by whatever particular route is convenient. [Wea49]

Common experience suggests that when a human attempts to translate a sentence from a language f to a language e, she does the following: 1) looks at the f sentence, 2) acquires its meaning, and 3) renders that meaning according to the rules of language e. There is no reason, in general, that computational approaches to natural language processing should blindly ape our intuitions about human language processing (indeed, we make no claims about the *psycholinguistic* plausibility of this work!), but this intuition does suggest that there may be subtleties of the translation process which are modeled poorly or not at all in the absence of semantic markings. From a perspective purely concerned with efficiency, if a multilingual translation system is to be constructed (and assuming an appropriate universal interlingua can be devised), the number of pairwise translation systems necessary will grow quadratically but the number of semantic parsers and generators only linearly in the number of languages.

One might reasonably object that a purely semantic model does not describe everything a human translator will do when asked to translate. Such a model should regard the translation of "gatos y perros" into "cats and dogs" or "dogs and cats" equally likely; a

[2] e.g. Star Trek's Universal Translator or the *Hitchhikers Guide*'s Babel Fish)

human, when asked to perform the same task, will almost invariably prefer "cats and dogs" as it preserves the word order of the original example. Other examples, preserving more complicated syntactic phenomena, are also possible.

It is undeniable that a good literary translation requires a level of fidelity to the source text which a semantic model will never capture—but one that, it must be admitted, even human translators are never able to model in full. More generally, demanding that our translation systems respect both syntax and semantics, when it's unclear that we know how to handle semantics alone, is too much. In this respect, the proposal of a semantic model of machine translation is a *relaxation* of the model putatively underlying modern translation systems. Let us worry about the difference between "cats and dogs" and "dogs and cats" once we have ensured that animal names are not rendered as something else entirely.

Moreover, the availability of a universal meaning representation as an intermediate form would allow a more seamless integration of various other tools (e.g. coreference resolvers) into the translation process. The final motivation of this work is the possibility of laying the groundwork for a machine translation system capable of employing reasoning.

The theoretical framework for semantics-based machine translation continued to attract attention long after Weaver's time. The Vauquois Triangle (Fig. 1.1) [Vau68] characterizes numerous families of techniques for machine translation, at increasing levels of abstraction. Vauquois predicts "direct", word-to-word translation, which may roughly be said to capture modern phrase-based methods; a "syntactic transfer" from parse trees in one language to parse trees in another, and finally "semantic transfer" and "interlingua" at the highest levels of the pyramid. Notably, Vauquois allows that semantic representations are not necessarily sufficient to serve as a universal interlingua, and considers the possibility of translation between multiple semantic representations as an alternative to the use of a true interlingua.

One reason for skepticism toward a universal semantic representation is provided by the aboriginal languages of Australia, many of which make exclusive use of absolute directional markers ("I am standing north of the house" rather than "in front of the house") [JT06]; in principle, any interlingua which through which we wish to adequately translate between one of these languages and a language with relative directional markers must encode both

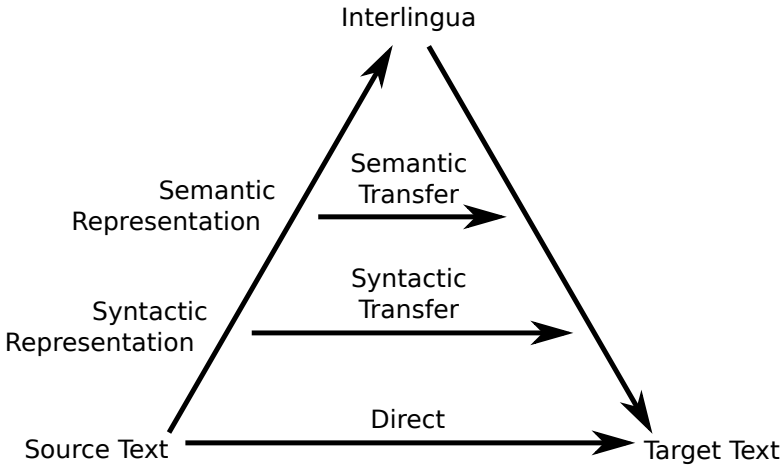

Figure 1.1: The Vauquois Triangle

absolute and relative position. If we attempt to design an interlingua for all of Earth's languages it will certainly be so weighed down by baggage of this kind that all parsing will be impossible.

Nevertheless we might reasonably expect to begin with a semantic representation which is sufficient to describe all relevant grammatical phenomena in two languages; we might further dispense with the syntactic segments of the triangle altogether, and parse into our universal semantic representation directly from raw text. This gives rise to a degenerate Vauquois triangle (Fig. 1.2), the model that we will consider for purposes of this thesis.

As discussed, the system presented here is not a high-quality, broad-coverage machine translation system, and is not intended to be. But I wish to demonstrate (and think I have succeeded in demonstrating) that, following very recent successes in analyzing into and realizing out of λ-calculus expressions, it is possible to do statistical, semantic machine translation in a principled and reasonably efficient way, and that this approach to translation is capable of outperforming naïve phrase-based systems. I hope (though here I am realistic about how much motivation a bachelor's thesis can provide) to motivate with this work further research on semantic translation systems, with the goal of eventually bringing them up to parity with (and beyond) phrase-based and syntax-based systems in coverage and generality.

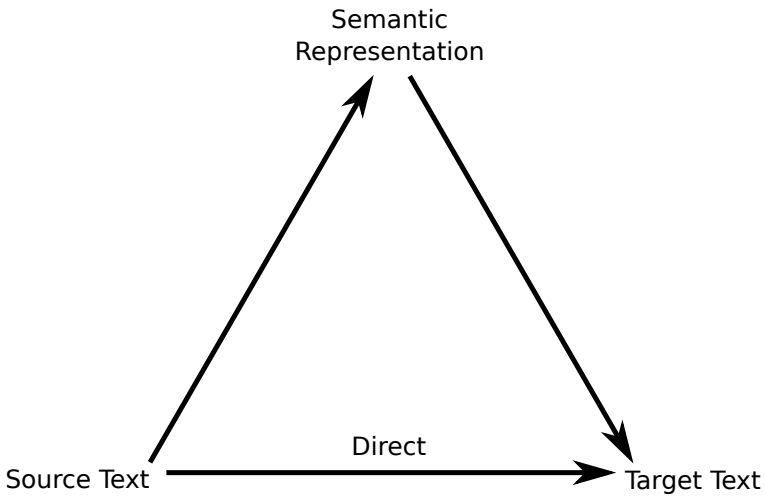

Figure 1.2: A degenerate "semantic Vauquois Triangle"

Chapter 2

Related Work

> Long, long ago there was a great rose of Sharon that counted eight thousand years as one spring and eight thousand years as one autumn. [...] Yet Pengzu alone is famous today for having lived a long time, and everybody tries to ape him.
>
> ZHUANGZI

2.1 Machine Translation

2.1.1 Interlingua

The very earliest successful attempts machine translation were built on large tables of hand-specified rules. These systems required large amounts of highly specialized human knowledge (and consequently great cost) to produce, but with the investment of enough effort could be coaxed into producing reasonable output. These systems were, with few exceptions, either *transfer-based* or *interlingual*. In transfer-based systems, sentences were analyzed in the source language, and rules specified transformations of that analysis to produce analyses of the target sentence, which could then be used to generate the target text. Interlingual systems, meanwhile, had as their underlying theoretical model precisely the one described in the introduction: The process of machine translation was all analysis

and no transfer, and it was sufficient to parse the source sentence into interlingua and then generate from interlingua into the target.

Why did such systems fail? Principally because systems dependent on the manual specification of rules were simply unable to keep up with the coverage of models that were learned from the enormous parallel corpora suddenly available. Systems requiring hand-specified rules are brittle, and do not adapt easily to changing patterns of language use or lend themselves to easy adaptation in highly technical or otherwise rarefied domains.

There is, nonetheless, a great deal to be learned from those early attempts at semantics-based machine translation, particularly with regard to the hard problem of interlingua design. One notable success in rule-based machine translation is SYSTRAN, which began developing a transfer system in 1968 as has continued to remain competitive to the present day; see [SDVB01] for a (comparatively) recent system. Much of the research challenge in better hand-constructing machine translation systems lies in simplifying the rule coding process and eliminating redundancy, and thus provides little insight into the process of translation itself; one reason to be skeptical of this general family of approaches is that SYSTRAN has invested many person-centuries of work in a system whose performance is matched by an out-of-the-box phrase-based system trainable in hours.

Work on interlingual machine translation has more or less ceased in recent years. One of the last successful such systems was KANT [MNC91], developed at Carnegie Mellon, which also relied principally on expert-constructed rules (although with partially automated lexicon building) and required a special kind of "controlled language" in the source text [MN94]. For a general survey of interlingual approaches to machine translation, see also [DHL04].

2.1.2 Statistical Transfer

Weaver again:

> I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.

Weaver prefigured the "statistical revolution" which occurred in machine translation in 1990 with the introduction of the so-called IBM Models 1-5 [BCP+90]. The IBM models, rather than attempting to provide any kind of linguistic account of translation, simply assume, in keeping with Weaver's intuition, that "French" is some statistical process which distorts English sentences, and attempts to recover the output. Formally, translation is treated as a "noisy channel" process; we wish to model the probability $p(e|f)$ of observing an English text e given a French text f, and to choose the e maximizing this probability. Bayes' rule tells us that this may be estimated as

$$p(e|f) \propto p(f|e)p(e)$$

(disregarding the usual denominator $p(f)$ on the right-hand side, as it will remain constant over all choices of translation). The first term, $p(f|e)$, is a translation model which describes the relationship between the two languages; the second, $p(e)$ is a language model which may be constructed from monolingual data.

But how to model $p(f|e)$? IBM's general observation was that it is possible to decompose this probability in terms of *word alignments* a between the two languages, and that the joint probability $p(f, a|e)$ is easier to model than $p(f|e)$ alone. Models 1 through 5 presented various ways of generating these alignments with increasing sophistication. Models 1 and 3, the most influential, will be discussed in more detail.

Model 1 is simple: It learns a one-to-one (but not necessarily onto) correspondence between the words of the source language and the target language. It models the joint probability of alignments and source sentences as

$$\begin{aligned} p(f, a|e) &= p(a|e)p(f|a, e) \\ &= \frac{1}{(l+1)^m} \prod_{i=1}^{m} p(f_i|e_{a_i}) \end{aligned}$$

Once this probability is known, the alignments can be marginalized out to recover the original translation probability

$$p(f|e) = \sum_a p(f, a|e) \quad .$$

Thus the only quantity that needs to be estimated in order for translation to work is the pairwise probability $p(f_i|e_j)$ of words. This can be estimated efficiently using expectation maximization, and is guaranteed to converge to the global optimum.

But this is a rather poor model: It forbids multi-word alignments and relies entirely on the language model to correctly determine the syntax of the target sentence. Model 3 attempts to rectify this problem by introducing a notion of "fertility" which allows one English word to map to multiple French words (so that French *phrases*, rather than single words, can determine translation), and adds a distortion model which captures word order.

This "statistical revolution" rapidly reached syntax-based machine translation [Wu97, Blo00, Chi05]. The key insight enabling the extension of statistical methods to syntax was the observation that the word alignments used by phrase-based systems could be used as a starting point for learning a synchronous context-free grammar in the source and target language; then, when a new sentence was observed in one language, its derivation on one side of the SCFG could be used to efficiently generate the translation on the target side.

In general, hierarchical translation systems have proven very successful in translating between language pairs with highly divergent syntax [ZVOP08], as they are more tolerant than phrase-based systems of long-range reorderings. The success of statistical syntax-based systems offers some evidence that as we ascend the Vauquois pyramid, richer descriptions of the relationship between source and target language permit more fluent and accurate translation.

But why not stop here? Just as there are phenomena (e.g. long-range reordering) easily captured by syntactic but not phrase-based systems, there may be *semantic* phenomena whose realization in source and target languages are so divergent they cannot be captured by an SCFG. As mentioned in the introduction, we would also like a unified model of meaning which allows us to easily incorporate automated reasoners and other tools already

designed to operate on logical forms into our translation systems.

2.2 Semantic Parsing

2.2.1 Classical

The earliest approaches to semantic parsing relied on hand-constructed rules rules. One successful example is Bobrow et al.'s GUS system [BKK+77]. GUS, a "frame-driven dialog system" used a chart parser with hand-constructed rules to extract meaning from input text; meaning itself was represented in a standard frame-semantic fashion. Notably, GUS already incorporated reasoning into its dialog with the user, transforming these frames into database queries which could then be reformulated as responses to the user. But semantic parsing with hand-constructed rules suffered from the same shortcomings as the hand-constructed machine translations described above, and the approach was not long-lived in the research community.

2.2.2 Statistical

More recent work has focused on statistical approaches to semantic parsing. Early work on learning variable-free logical expressions includes that of Ge and Mooney [GM06] who construct "semantically augmented parse trees," extending a Collins parser to place semantic labels on each node, Kate and Mooney [KM06] which learns a support vector machine with a string similarity kernel for each production in a specially-constructed meaning representation language, Wong and Mooney's WASP [WM06] which learns a synchronous context free grammar over logical expressions and natural language representations, and Lu et al. [LNLZ08] who learn a hybrid tree simultaneously encoding syntax, semantics and sentence surface form.

The WASP projects, in particular, are dependent on the particular choice of variable-free logical expressions as meaning representation; the work was eventually extended to λ-calculus expressions in 2007 [WM07a] in the form of λ-WASP, which permits representation of λ-calculus expressions in the formulation of their CCG.

Some recent work also studies the problem of unsupervised semantic parsing, in cases

where semantically-annotated sentences are not available; this includes Poon & Domingos, who learn a Markov logic network in order to induce semantic forms [PD09]; Artzi et al.'s semantic parser bootstrapped from transcripts of interactions with a dialog system [AZ11]; and Liang's work on learning semantic representations from question-answer pairs [LJK11].

This parser used in this work is the one introduced by Kwiatkowski et al. [KZGS10] in 2010, which learns a combinatory categorial grammar annotated with λ-calculus expressions, providing a mapping from natural language sentences into λ-calculus representations (and, though they do not explore it in that paper, potentially vice-versa). Several variations on that work have been proposed, including templating [KZGS11] and inclusion of context [ZC09] (though this last paper relied upon a set of hand-engineered templates which Kwiatkowski et al., and the present work, do not).

CCGs more generally have been used for a variety of problems in semantics, including child language acquisition [Ste96] and "wide-coverage" semantic parsers [BCS+04].

2.3 Semantic Generation

2.3.1 Classical

The opposite of the problem provided by semantic parsing—given some meaning representation, how is it realized as a natural language string?—has long been recognized as closely related to the parsing problem. Early natural language generation systems, like their counterparts in analysis, tended to be based primarily in hand-constructed rules (see [TSRD06] for an example). Classical generation of this kind specifically targeted at performing natural-language realization out of logical forms includes [Wan80], which describes hand-constructed rules for transforming lambda expressions into constituency parses, and from there to fully-realized sentences. The structure search problem discussed in that paper is very similar to the one considered in generation here; as will be seen, it is possible overcome some of the combinatorial obstacles encountered there with standard heuristics. This family of approaches to generation, like the analogous parsing problem, tends to have low coverage.

2.3.2 Statistical

An recent example of the approach undertaken in this thesis is given in [WM07b]. They take a acquired semantic parser in their WASP system, and use the parser's learned SCFG with a standard chart-based generation algorithm to perform natural language realization. Wong and Mooney note that this general strategy of modeling semantic parsing and tactical generation as the same problem is in fact an old one, dating back to 1975 [Kay75] and that in general parsing charts may also be used in generation algorithms. Chart generation for CCG in particular includes the work of White and Baldridge [WB03].

The only recent work we are aware of focused specifically on the task of performing semantic generation out of lambda calculus expressions is that of Lu and Ng [LN11], who learn a "lambda-hybrid grammar" and, like Wong and Mooney, use a synchronous context-free grammar as the general model for generation. That work, however, does not come paired with a parsing algorithm; it is desirable, for the joint task of parsing and generation, to have some symmetry between the structures learned by our parser and our generator (and, following Wong and Mooney, to perhaps share data structures). Thus the new generation algorithms presented in this thesis differ from the work of Lu in the use of a CCG, and generally in their close relationship with the parser described by Kwiatkowski et al.

Chapter 3

Preliminaries

> [...] and if you are going a thousand *li*, you must start getting the provisions together three months in advance.
>
> ZHUANGZI

I begin with a presentation of the basic formal tools used to develop the parsing and generation algorithms used in this thesis.

3.1 CCG

Both the semantic parsing and generation algorithms presented in this thesis learn a combinatory categorial grammar, or CCG. Combinatory categorial grammar [Ste00] is a mildly context-sensitive grammar formalism [VSW94] (of equivalent power to linear indexed grammars, tree adjoining grammars and head grammars), attractive for various reasons: It is expressive, naturally interpretable in a logical setting (CCG categories can be made to map directly onto lambda calculus types), efficiently parsable, and perhaps psycholinguistically plausible [RHK06].

A CCG consists of a language-specific lexicon Λ, and a set of universal rules which describe how items in the lexicon may be combined in order to form sentences of the language. Every entry in the a CCG lexicon is a pair $\alpha : t$, where α is a string in the language and t is a *syntactic category*. Categories are either primitive or complex; the

number and naming of primitive categories depends on the construction of a given lexicon.

Strings combine to form new strings according to their categories. In this thesis a slightly restricted set of combinators is considered in order to ensure computational efficiency (and hopefully not at the expense of expressive power). The combinators used are described below. First are the application combinators:

$$\frac{\alpha : X/Y \quad \beta : Y}{\alpha\beta : X} > \qquad \frac{\beta : Y \quad \alpha : X\backslash Y}{\beta\alpha : X} <$$

Intuitively, these allow strings α with type X/Y or $X\backslash Y$ to "look" for arguments of type Y, giving back strings of type X. X/Y looks to the right, and $X\backslash Y$ looks to the right. `brown` : N/N might combine with `dog` : N to give `brown dog` : N, or alternatively `dog` : $N\backslash Adj$ combine with `purple` : Adj to give `purple dog`. Next are the composition combinators:

$$\frac{\alpha : X/Y \quad \beta : Y/Z}{\alpha\beta : X/Z} B_{>} \qquad \frac{\beta : Y\backslash Z \quad \alpha : X\backslash Y}{\beta\alpha : X\backslash Z} B_{<}$$

These correspond to functional composition in the same way that the application combinators correspond to functional application, changing the "return type" but not the type of the argument. Continuing with the previous example, with `big` : $Adj\backslash Adv$ in the lexicon, it is possible to produce `big dog` : $N\backslash Adv$.

Most formulations of CCG also include a type-raising combinator

$$\frac{\alpha : X}{\alpha : T/(T\backslash X)}$$

(and a corresponding backwards combinator). For purposes of computational efficiency the raising combinator is omitted in this work; empirical results suggest that the omission does not render the parsing task impossible, and most linguistic phenomena that would ordinarily require a raising combinator are simply described in some other way in the lexicon.

3.2 λ-calculus

The λ-calculus, first introduced by Church in [Chu32], has long been a popular choice for abstract meaning representation in computer systems. It is an expressive (indeed,

Turing-complete) programming language syntax, but also unambiguous, easily parsable and straightforward to manipulate computationally.

This thesis deals with *typed* λ-calculus expressions, in which every primitive object is labeled with one of e (an entity), i (a number) or t (a truth value); and every function which takes as input an object of type x and returns an object of type y has type $\langle x, y \rangle$. The λ-calculus expressions in the training data make use of a set of named primitives, both entities and functions, whose type signatures are assumed to be known in advance.

The algorithms in this thesis rely in particular on the ability to solve a series of higher-order unification problems; that is, given a function h, to find pairs of functions f and g such that $(fg) = h$ or $(\lambda x.(f(gx))) = h$. This process is known, in general, to be undecidable [Hue75]; even with some limitations it is possible that the number of pairs f and g is exponential in the length of h, so a moderately restricted form must be considered in order to constrain the available splits. Following [KZGS10], the following restrictions apply:

1. **No vacuous variables**: neither f nor g may contain arguments that appear in their bodies (as will be seen, this corresponds to an elimination of the raising combinator in Sec. 3.1).

2. **Limited coordination**: fix some N; then the expression g may not contain more than N conjuncts (or disjuncts) appearing in h. If coordination is unrestricted, any subset of the conjuncts may be selected; these subsets grow exponentially in the size of the expression. Like [KZGS10], choose $N = 4$.

3. **Limited application**: f may not contain any new variables applied to non-variable subexpressions in h.

With all of these restrictions, the number of splits for any expression will be at most an Nth-degree polynomial in the length of the expression.

Note that the two conditions for splitting ((fg) and $(\lambda x.(f(gx)))$) correspond directly to the two combinators introduced in the CCG framework, and that types of the lambda calculus can be made to correspond directly to CCG categories with an appropriate choice

of basic categories. This leads directly to the development of the joint λ-calculus and CCG splitting process which drives the algorithms presented here.

3.3 The joint splitting process

Like much of the related work on semantic parsing and generation, the central structure used in these experiments is a hybrid language-meaning parse tree, in which every node is labeled with both a CCG category, a lambda-calculus expression, and a natural language string, and in which the root node contains both the complete semantic representation and the full natural language text, and leaf nodes correspond to entries in a hybrid CCG lexicon, analogous to the an ordinary CCG lexicon but with the addition of lambda calculus expressions for each entry.

We may then think of the process of parsing as attempting to reconstruct this tree observing only the text at the leaves, and the process of generation as finding the tree observing only the lambda calculus expression at the root. The discovery of this tree relies upon the splitting process for lambda calculus expressions described in the preceding section.

Formally, each node of the tree is a triple $\alpha : t \vdash e$, with α a string, t a CCG type and e a λ-expression. This gives rise to the following inference rules for paired language-meaning representations:

$$\frac{\alpha : T(f(g))/T(g) \vdash f \quad \beta : T(g) \vdash g}{\alpha\beta : T(f(g)) \vdash f(g)} >$$

$$\frac{\beta : T(g) \vdash g \quad \alpha : T(f(g))\backslash T(g) \vdash f}{\beta\alpha : T(f(g)) \vdash f(g)} <$$

$$\frac{\alpha : T(\lambda x.f(g(x)))/T(g) \vdash f \quad \beta : T(g) \vdash g}{\alpha\beta : T(\lambda x.f(g(x))) \vdash \lambda x.f(g(x))} B_>$$

$$\frac{\beta : T(g) \vdash g \quad \alpha : T(\lambda x.f(g(x))\backslash T(g)) \vdash f}{\beta\alpha : T(\lambda x.f(g(x)) \vdash \lambda x.f(g(x))} B_<$$

Where the CCG type inference function T is defined as follows:

$$T(x) = \begin{cases} NP & x = e \\ S & x = t \\ T(y)|T(z) & x = \langle y, z \rangle \end{cases}$$

Note that in the datasets considered for this experiment, the numeric type i will never appear in the signature of the parse of a sentence. In general, this formalism results in a rather unusual-looking set of CCG types; all proper nouns will have type N, but other nouns will often be assigned types like $S|NP$, corresponding to their semantic representation as single-argument predicates (e.g. the *state* function) rather than atomic entities.

3.4 Corpus

The data used for the experiments presented in this thesis are drawn from the GeoQuery corpus [ZM96]. The GeoQuery dataset consists of 880 logical expressions expressions paired with English-language representations. The questions were generated by asking undergraduates at UT Austin (and later, users from the web) to produce English queries for a database of geographic facts; these queries were then manually translated into logical representations. Two forms, both a lambda-calculus and variable-free representation of the logical representations, are available in the GeoQuery dataset; this thesis presents results only for the lambda calculus expressions.

The first 250 sentences in the corpus, which will be referred to as the Geo250 dataset, were also translated from English into Spanish, Turkish and Japanese; these sentences are used for translation experiments. The complete corpus, available only in English (and, when understood as such, referred to as the Geo880 corpus) was used in addition to the Geo250 dataset for parsing and generation experiments.

Note here Kate's observation that the examples in the GeoQuery dataset are, in general "harder" than many of the other standard datasets used in semantic parsing like ATIS and RoboCup [Kat07]—the GeoQuery entries tend to involve deep nesting (e.g. "which states have points higher than the highest point in Colorado"). The average sentence length in

Sentence	λ-expression
how many colorado rivers are there	`(count $0 (and (named:t $0 colorado:n) (river:t $0)))`
what is the population of hawaii	`(population:i hawaii:s)`
what is the length of the mississippi river	`(len:i mississippi_river:r)`
what length is the mississippi	`(len:i mississippi_river:r)`
colorado nun cak tane nehri vardir	`(count $0 (and (river:t $0) (loc:t $0 colorado:s)))`

Figure 3.1: Sample entries from the GeoQuery dataset

Geo880 is 7.48 words, and the average lambda-calculus expression contains 6.47 tokens.

Throughout this thesis, the results of experiments on the complete Geo880 data set are based on a single train-tune-test split. The testing data are identical to the conventional 280-sentence test set for the Geo880 corpus, while the training and tuning split is created by setting the final 100 sentences of the standard training set aside for tuning (the same split used by Lu and Ng). The results of experiments on the Geo250 dataset are the average of ten-fold cross-validation, using the cross-validation split in run-0 of [KZGS10].

Chapter 4

Parsing

> Don't listen with your ears, listen with your mind. No, don't listen with your mind, listen with your spirit. Listening stops with the ears, the mind stops with recognition.
>
> ZHUANGZI

Our discussion of the system presented in this thesis begins with the first stage: the parser. Semantic parsing, as has been discussed, has various applications in its own right; more importantly, a robust semantic parser is crucial as the first stage of analysis in semantics-based machine translation. While contributions to the parsing problem are less substantial than the other work presented in this thesis, this section describes some new results with an extended parser feature set, and discuss the results of the baseline parser in order to provide context for the interpretation of parser output in the complete translation pipeline.

4.1 The Parsing Model

The first parser discussed is identical to the parser, already mentioned several times, introduced by Kwiatkowski et al. in 2010 [KZGS10].[1] The reader is referred to that paper for a more detailed exposition of the parser's inner workings. The training algorithm is given as

[1] The authors of that paper generously provided their code for use in these experiments.

Algorithm 1, and a subroutine for expanding the lexicon in Algorithm 2.

A log-linear model on productions assigns probabilities to parse trees; the joint probability of a parse y and a logical representation z conditioned on a sentence x is given by

$$p(y, z|x; \theta, \Lambda) = \frac{\exp(\theta \cdot \phi(x, y, z))}{\sum_{(y', z')} \exp(\theta \cdot \phi(x, y', z'))} \quad ,$$

given a feature vector ϕ to be defined shortly. Finding the most likely parse for a given sentence x is then simply the problem of calculating

$$\arg\max_{z} p(z|x_i; \theta, \Lambda) \quad ,$$

with

$$p(z|x_i; \theta, \Lambda) = \sum_{y} p(y, z|x_i; \theta, \Lambda) \quad ,$$

which, if all features are local, can be approximated with a pruned chart parser. ϕ is optimized using stochastic gradient descent; the update to the objective O is given by

$$\frac{\partial O_i}{\partial \theta_j} = \Delta = \mathrm{E}_{p(y|x_i, z_i; \theta, \Lambda)}[\phi_j(x_i, y, z_i)] - \mathrm{E}_{p(y, z|x_i; \theta, \Lambda)}[\phi_j(x_i, y, z)] \quad .$$

The algorithm takes as input a pair of n training examples, consisting of a paired natural-language sentence and λ-calculus expression. The lexicon Λ is initialized to contain both these training examples (with syntactic type S), and a list of proper nouns and their corresponding semantic tokens (with syntactic type NP). When new lexical features are added to the parameter vector θ, they are initialized according to co-occurrence statistics calculated with a standard aligner (see the original paper for details).

At a high level, this is simply a structured perceptron. At each step, it determines what lexicon update will best best improve the current parse, and then refines the parameters of the model correspondingly. Note that in general, splitting would tend to decrease the likelihood of the data; however, because positive weights are assigned to newly introduced lexical features the algorithm will (initially) favor splitting over merging. To calculate the stochastic gradient update parameters $\mathrm{E}_{p(y|x_i, z_i; \theta, \Lambda)}[\phi(x_i, y, z_i)]$ and $\mathrm{E}_{p(y, z|x_i; \theta, \Lambda)}[\phi(x_i, y, z)]$, the inside-outside algorithm is used.

Algorithm 1 TRAIN-PARSER

for $t = 1..T$ **do**
  for $i = 1..n$ **do**
    $y^* = \arg\max_y p(y|x_i, z_i; \theta, \Lambda)$
    $\Lambda \leftarrow \Lambda \cup \text{NEW-LEX}(y^*)$
    $\gamma = \frac{\alpha}{1+c(i+tn)}$
    $\Delta = \mathrm{E}_{p(y|x_i,z_i;\theta,\Lambda)}[\phi(x_i, y, z_i)] - \mathrm{E}_{p(y,z|x_i;\theta,\Lambda)}[\phi(x_i, y, z)]$
    $\theta \leftarrow \theta + \gamma\Delta$
  end for
end for

Algorithm 2 NEW-LEX

L_1 = all lexicon entries obtained by merging a node in y
L_2 = all pairs of lexicon entries obtained by splitting a node in y
$L = L_1 \cup L_2$
$l^* = \arg\max_{l \in L} p(y^*|x_i, z_i; \theta', \Lambda \cup l) - \max_y p(y|x_i, z_i; \theta', \Lambda \cup l)$
return l

4.1.1 Baseline Features

The parser makes use of two kinds of indicator features: lexical and semantic. Each entry in the lexicon has a corresponding feature that triggers when it is used in a sentence; each pair of a named function and primitive in the logical representation have a feature that triggers when the primitive appears as the ith argument to the function. This baseline parser is referred to as KZGS in the table of results below.

4.1.2 Categorial Features

Note that all of the features used in Kwiatkowski et al.'s parser occur at extremal nodes of the tree—either the root (semantic features) or the leaves (lexical features). I extend the KZGS parser by adding a third set of features which trigger on internal nodes of the tree according to their CCG categories. In particular, every split $[\alpha\beta : X] \rightarrow [\alpha : X|Y][\beta : Y]$ will trigger an indicator feature $[X \rightarrow X|Y, Y]$. For the sake of generality, all directional information contained in the syntactic category is discarded, so e.g. the feature component associated with the type $S/NP\backslash(NP/NP)$ is $S|NP|(NP|NP)$. This feature set is referred to as KZGS+CCG.

While this feature is principally motivated by the generation problem (and discussed for

	KZGS			KZGS+CCG		
	Recall	Precision	f_1	Recall	Precision	f_1
EN880	0.850	0.967	0.905	0.807	0.915	0.857
EN250	0.768	0.924	0.837	0.776	0.949	0.851
ES250	0.780	0.943	0.852	0.780	0.943	0.852
TR250	0.664	0.929	0.771	0.652	0.913	0.758
JA250	0.796	0.921	0.853	0.768	0.881	0.820

Table 4.1: Results for basic parser with baseline and augmented feature sets.

	KZGS			KZGS+CCG		
	Recall	Precision	f_1	Recall	Precision	f_1
EN880	0.889	0.896	0.892	0.817	0.820	0.819
EN250	0.812	0.825	0.818	0.832	0.839	0.835
ES250	0.832	0.851	0.841	0.836	0.852	0.844
TR250	0.716	0.752	0.733	0.684	0.724	0.703
JA250	0.824	0.824	0.824	0.808	0.808	0.808

Table 4.2: Results for skipping parser with baseline and augmented feature sets.

that purpose in more detail in Chapter 5) we initially believed that it might also help in semantic parsing for the same reason features triggered on internal nodes in discriminative constituency parsing models are useful. The evaluation results do not support this intuition; the few experiments in which KZGS+CCG scores higher are not statistically significant.

4.2 Evaluation

The parser fails to produce a semantic representation on a nontrivial fraction of sentences: 17% for English and Spanish, 28% for Turkish and 14% for Japanese. This fact presents a serious obstacle for a potential translation system: For a substantial portion of the input, it will be impossible to translate at all! Falling back on the skipping parser might help some; as expected, allowing word-skipping greatly increases recall at the expense of precision. Using the skipping parser as a fallback, however, also introduces a risk of undergenerating on the target side and losing content from the sentence. Because I envision the translation system from this thesis used (if it is used in practice) in conjunction with a phrase-based

system as a last-resort fallback, the skipping parser is omitted from the translation pipeline entirely: It is less trustworthy state-of-the-art PSMT. The generally high quality of these results on successful parses, however, suggests that not too much noise will be introduced into the translation process by the parser, and that the KZGS parser will be an adequate first stage in the translation pipeline.

Chapter 5

Generation

> Saying is not blowing breath, saying says something; the only trouble is that what it says is never fixed. Do we really say something? Or have we never said anything?
>
> ZHUANGZI

This section introduces experiments on the generation side only, designed to provide some characterization of generator behavior when decoupled from noise (or empty parses) output by the parser. Rather than simply plugging in a state-of-the-art lambda-calculus generator like that of [LN11], I have essentially inverted the parser described in Chapter 4, with the intuition that the comparatively rich feature model used there will also result in a good generation algorithm, and in the hopes of eventually enabling joint training for parsing and generation. This symmetry between the parser and generator allow us to introduce a very simple, PCFG-driven generator which actually performs quite well and requires no retraining beyond the training of the parser. That generation algorithm, as well as one which more closely mirrors the discriminative training for the parser, are presented below.[1]

[1] The generation code used for the experiments presented here relies on the implementation of the splitting algorithm provided by [KZGS10], described in the previous section.

5.1 The Generation Model

Both generation models introduced rely upon the creation of a hypergraph describing all possible derivations of the target sentence. An approach very similar to the one in [Chi05] is used to heuristically find the single derivation in this hypergraph scored highest by both the edge weights and a language model.

The generation of the hypergraph itself presents several challenges. Unlike the analogous parsing problem, there is no constraint on the overall length of the generated sentence. Many of the lexical entries produced by the learning algorithm are totally vacuous, and can be chained indefinitely to produce sentences of infinite length. (For example, `what` : $S/NP \vdash \lambda x.x.$) To restrict the size of the hypergraph generated before realization, we take an iterative deepening approach to this problem, passing as an argument to the tree generation algorithm a maximum depth which is increased until a tree containing an acceptable solution is discovered. This algorithm is given as BUILD-HYPERGRAPH (Algorithm 3).

The two algorithms presented differ only in the technique used to acquire weights for the hypergraph edges. For all of the experiments below, a graph of splits is generated, and weights assigned to its hyperedges using various strategies. Finally, a standard cube-pruning decoder is used to intersect the hypergraph with a language model to produce final sentences.[2] For all of the experiments below, the language model employed is trained on the same dataset as the generator.

5.1.1 PCFG Generation

The simpler of the two generation approaches (referred to as "Model 1" below) uses only output from the parser training process, and does not require retraining for generation. It learns a generative model: Leaf nodes of the hypergraph are identified by matching them against the lexicon output by the parser training process, and edge weights are assigned by learning a PCFG over CCG categories using the final parses of the training data.

Concretely, the model learns a PCFG where the weight of each production is determined by three factored probabilities: a categorial probability p_1, a semantic probability p_2 and a

[2] The cube pruning implementation in Alexander Rush's Scarab package was used in this step.

Algorithm 3 BUILD-HYPERGRAPH

Globals: M (visited nodes), N (reached nodes), E (edges), D (max depth), Λ (lexicon)
Arguments: $n = t \vdash e$ (current node), d (current depth)
if $n \in M$ **then**
    return $n \in N$
end if
$M \leftarrow M \cup \{n\}$
$L = \{(\alpha' : t' \vdash s') \in \Lambda : t = t' \wedge s = s'\}$
if $|L| > 0$ **then**
    BUILD-TERMINAL(n)
    return `true`
end if
if $d = D$ **then**
    return `false`
end if
return BUILD-NONTERMINAL(n, d)

Algorithm 4 BUILD-TERMINAL

Globals as above
Arguments: $n = t \vdash e$ (current node)
for all $(\alpha' : t' \vdash s') \in L$ **do**
    $e = ()$
    for all $w \in$ TOKENIZE(α') **do**
        APPEND(e, w)
    end for
    $E \leftarrow E \cup \{(n, e)\}$
end for
$N = N \cup \{n\}$

Algorithm 5 BUILD-NONTERMINAL

```
Globals as above
Arguments: n = t ⊢ e (current node), d (current depth)
k = false
for all (p, q) ∈ SPLITS(n) do
    if ¬BUILD-HYPERGRAPH(p, d + 1) then
        continue
    end if
    if ¬BUILD-HYPERGRAPH(q, d + 1) then
        continue
    end if
    k = true
end for
if k then
    N = N ∪ {n}
end if
return k
```

lexical probability p_3:

$$\begin{aligned} p((\alpha : T \vdash a),\ (\beta : U \vdash b) \rightarrow (\gamma : V \vdash c)) \ &= \lambda_1 \cdot p_1(T, U|V) \cdot \\ &\quad \lambda_2 \cdot p_2(a, b|c) \cdot \\ &\quad \lambda_3 \cdot p_3(\alpha, \beta|T, U, a, b) \quad . \end{aligned}$$

The categorial probability p_3 models the probability of observing a split $X|Y, Y \rightarrow X$ given a parent category X. The semantic probability p_3 models the probability of observing a split $f, g \rightarrow h$ (where $h = f(g)$ or $\lambda x.f(g(x))$. Finally, the lexical probability, assigned only to edges incident on leaves (and set to 1 elsewhere), models the probability that an expression a with syntactic type U will produce a terminal string α.

We may obtain maximum-likelihood estimates for these probabilities by simply counting the number of times each production occurs in the parse trees for all the training sentences given as output by the parser training process, and use Laplace smoothing to account for previously-unseen productions. For the GEO880 dataset, weights are assigned to the three λ_i, and an additional language model probability, using minimum error rate training [Och03] on a held-out set of 100 examples; for the GEO250 dataset, weights for categorial, semantic, lexical and LM probabilities are set at $0.05, 0.02, 0.05$ and 1 respectively.

5.1.2 Log-linear Generation

The next decoder ("Model 2") is a more faithful inversion of the parsing algorithm. Recall that in the parser we wished to find the most probable parse conditioned on the observed text; here we do the opposite, i.e. find

$$\arg\max_x p(x|z;\theta,\Lambda)$$

with

$$p(x|z_i;\theta,\Lambda) = \sum_y p(y,x|z_i;\theta,\Lambda)$$

for a sentence x and a logical form z. As a consequence the training algorithm is only subtly different from the training algorithm (Algorithm 1) employed for parsing. For the sake of brevity the complete parser training algorithm is omitted; instead simply note that the new stochastic gradient update is given by

$$\frac{\partial O_i}{\partial \theta_j} = \Delta = \mathrm{E}_{p(x|y_i,z_i;\theta,\Lambda)}[\phi_j(x,y_i,z_i)] - \mathrm{E}_{p(x,z|y_i;\theta,\Lambda)}[\phi_j(x,y_i,z)] \quad ,$$

again changing the place of x and z.

The weight vector over features given by θ is used to assign weights to each hyperedge of the split graph during the training process, and eventually during decoding. As before, the heuristic cube-pruning algorithm is used to calculate the arg max over all derivations in order to incorporate a language model. The problem of finding expected feature values is simply a special case of the standard outside algorithm for applying expectation semirings to hypergraphs [LE09], and solutions may be efficiently computed.

5.2 Evaluation

This section presents both standard BLEU scores, and a "trimmed" BLEU score which does not penalize the generator for sentences on which it fails to generate. For reference, the scores achieved by Lu and Ng's generator are also provided in the LN column.

	Model 1		Model 2		LN
	base	trim	base	trim	base
EN880	44.58	46.68	43.61	46.25	54.58
EN250	58.67	54.66	58.30	55.89	–
ES250	59.08	64.16	59.08	65.17	–
TR250	32.97	36.55	30.32	33.79	–
JA250	52.96	60.15	52.93	59.86	–

Table 5.1: Results for generation into English.

5.3 Discussion

All of the translation models presented fall substantially short of Lu and Ng's generator. An inspection of the translation results suggests that this is primarily a problem of brevity: our system tends to prefer, for example "population of denver" to "what is the population of denver", or "what states does the mississippi" to "through what states does the mississippi run". This is a natural consequence of using an inverted CCG parser for generation: the grammar learned is very good at identifying (and discarding) semantically redundant content in a sentence, but less equipped to identify syntactically necessary features. Nevertheless these models are interesting both as a new application of CCGs for generation out of logical forms, and as a potential starting point for a joint parsing/generation model for machine translation.

Chapter 6

Translation

> Master Lai grew ill. Gasping and wheezing, he lay at the point of death [...] Master Li, who had come to ask how he was, said, "Shoo! Get back! Don't disturb the process of change."
>
> ZHUANGZI

Having determined both how to parse into λ-calculus expressions and to generate out of them, we are ready to join the two systems together to create a complete translation pipeline. As pointed out before, there are circumstances under which both the parser and generator can fail to produce output, which means that this will not be a full-coverage translation system on its own; Chapter 4 already discussed a mechanism to remedy this problem by falling back on a phrase-based system when parsing or generation fails. This section focuses on the output of the pure semantic translation system without coupling, in order to characterize both the severity of empty output problem and the quality of the successful translations.

6.1 The translation models

The basic approach underlying both sets of translation experiments presented here is the same: Join together the semantic parser and semantic generator described in the preceding sections, passing the logical representation output by one as input to the other. I use

the KZGS parser from Chapter 4, and the Model 1 generator from Chapter 5. In order to avoid writing "KZGS–Model 1" throughout the rest of this thesis, I christen the resulting system BAST (perhaps "Born Again Semantic Translator"?), in keeping with a longstanding Egyptological naming convention in machine translation.

Evaluation, however, is complicated for a few reasons. Firstly, there is an asymmetry in the data required: As often happens in machine translation, large monolingual resources for the target language can prove useful for training a language model, and in this case, the entire generation model. Secondly, as discussed in previous sections, the parser simply fails to produce any output at all on a nontrivial fraction of input sentences, and in these cases translation is impossible. As a consequence, this section presents three evaluation methods which provide different insights into the quality and coverage of the translations produced.

The evaluations presented here describe a "balanced" resource scenario, in which the training data consist of a three-way parallel corpus of source sentences, target sentences and semantic representations. For this evaluation, all training data are drawn from the translated GEO250 dataset. Evaluation results for all languages are presented.

6.2 Evaluation

To characterize the circumstances under which the parser fails to generate, both "base" and "trim" results (analogous to the previous section) are given, providing a comparison of BLEU scores over both the unfiltered document (usually incurring a substantial brevity penalty) and BLEU scores over a filtered set of references which evaluate the precision of the successful outputs only. Table 6.2 shows automated evaluation of the balanced experiment. Manual evaluation of a limited subset of results from the balanced experiment is shown in in Table 6.6. Evaluators were asked to perform a binary assessment for both adequacy and fluency; for each output they were asked "Does this sentence adequately capture the meaning of the reference sentence?" or "Is this a fluent English sentence?" respectively. All evaluators were Columbia undergraduates and native English speakers.

For table entries marked with a single asterisk*, BAST outperforms the baseline with $p \leq 0.05$; with a double asterisk**, $p \leq 0.01$. Paired bootstrap resampling [Koe04] was

	EN	ES	TR	JA
EN	–	46.77 / 83.15	24.71 / 44.07	39.11 / 63.82
ES	44.28 / 80.84	–	24.82 / 45.19	42.18 / 64.66
TR	39.00 / 63.51	40.74 / 68.95	–	35.37 / 73.43
JA	42.48 / 45.07	44.54 / 51.10	27.81 / 53.12	–

Table 6.1: BLEU scores for translation. (BAST / Moses, untrimmed)

	EN	ES	TR	JA
EN	–	64.51 / 89.15	35.97 / 53.10	61.43 / 70.73
ES	57.74 / 84.88	–	35.25 / 52.82	60.63 / 71.17
TR	61.61 / 78.92	66.83 / 85.89	–	67.51 / 87.87
JA	**59.12*** / 52.27	**62.42*** / 57.07	38.81 / 62.54	–

Table 6.2: BLEU scores for translation. (BAST / Moses, trim)

used to estimate the significance of BLEU scores, and the sign test was used for manual evaluations.

6.3 Discussion

Observe that BAST significantly outperforms the phrase-based system on automated evaluation of Japanese-English translation and Japanese-Spanish translation, but underperforms the phrase-based system on Spanish-English translation. Additionally, manual evaluation demonstrates that the BAST's output is substantially more fluent, and perhaps more adequate, than the output from Moses.

Note that both Turkish and Japanese are substantially more remote syntactically from English (both Japanese and Turkish have Subject-Object-Verb order, while English and

	EN	ES	TR	JA
EN	–	.9518	.9155	.9536
ES	.9493	–	.8951	.9480
TR	.9714	.9777	–	.9650
JA	.9471	.9455	.9188	–

Table 6.3: BLEU brevity penalties (BAST, trim)

	EN	ES	TR	JA
EN	–	81.93 / 95.17	71.21 / 83.72	85.03 / 88.39
ES	79.01 / 94.31	–	71.80 / 94.50	84.55 / 91.91
TR	78.50 / 92.56	96.60 / 98.30	–	87.27 / 94.44
JA	78.81 / 84.26	79.58 / 84.17	71.64 / 82.43	–

Table 6.4: BLEU 1-gram scores for translation. (BAST / Moses, trim)

	EN	ES	TR	JA
EN	–	59.28 / 85.63	25.89 / 36.86	53.23 / 60.06
ES	47.86 / 78.27	–	26.07 / 35.78	53.23 / 59.81
TR	52.73 / 72.46	61.52 / 80.21	–	61.21 / 86.11
JA	51.21 / 31.29	58.61 / 38.75	29.41 / 51.89	–

Table 6.5: BLEU 4-gram scores for translation. (BAST / Moses, trim)

	Adequate		Fluent	
	E1	E2	E3	E4
TR-EN	88.6 / 85.8	**89.2*** / 82.4	**84.1**** / 72.7	**87.5**** / 73.3

Table 6.6: Manual evaluation results (BAST / Moses, trim). Numbers given are the percentage of sentences that were judged to be adequate and fluent respectively.

Input (JA)	mishigan ni rinsetsu suru shuu wa dochira desu ka
Reference	what state borders michigan
Moses	michigan state borders is the
BAST	which states border michigan
Input (JA)	kororado kawa wa ikutsu no shuu wo nagarete imasu ka
Reference	how many states does the colorado river run through
Moses	the colorado river run through how many states does the
BAST	how many states does the colorado river run through
Input (TR)	baskenti atlanta olan eyalete komsu eyaletlerden gecen nehir nedir
Reference	what rivers run through the states that border the state with the capital atlanta
Moses	capital atlanta OOV OOV states that OOV what is the river
BAST	*no output*
Input (TR)	spokane washington da ne kadar insan yasamaktadir
Reference	how many people live in spokane washington
Moses	spokane how many people live in washington
BAST	how many people live in spokane
Input (ES)	que es la poblacion de utah
Reference	what is the population of utah
Moses	what is the population of utah
BAST	how many people stay in utah
Input (ES)	que rios corren por los estados que bordean a el estado con la capital atlanta
Reference	what rivers run through the states that border the state with the capital atlanta
Moses	what rivers run through states that border the state with the capital atlanta
BAST	which rivers run through states bordering the state with the capital atlanta

Figure 6.1: Sample output from the translation system

Spanish have Subject-Verb-Object order); I hypothesize that the translation model has learned to capture long-range reordering phenomena which the phrase-based system has not. A detailed analysis of the n-gram-level BLEU scores (Tables 6.4 and 6.5) confirm the observation, evident from an inspection of the system's output (Table 6.1), that BAST's advantage over Moses, where present, is grammatical rather than lexical—it is no better at choosing unigrams, but is more often able to arrange them into correct 4-grams than is Moses. The BLEU brevity penalty is also revealing: A large part of BAST's poor performance against BLEU is attributable to undergeneration, a problem which may be fixed in future versions with the addition of an insertion bonus.

The substantial divergence in the automated and manual evaluation results for Turkish suggest that BLEU may be a poor proxy for the adequacy and fluency questions with regard to this system. BAST will translate "cuantas personas viven en los angeles" and "que es la poblacion de los angeles" identically, because their semantic representations are both `(population:i los_angeles:c)`; if the reference translator rendered the first as "what is the population of los angeles" and the second as "how many people live in los angeles", BLEU will heavily penalize at least one of BAST's outputs, even if users of the translation system are unconcerned with the difference between the two. In general these results provide confirm that in cases where fidelity to the particular word choice of the source sentence is important, phrase-based approaches are preferable. Conversely, when fluency and semantic adequacy are more important, semantic translation is sometimes superior.

Chapter 7

Conclusion

> Though the grease burns out of the torch, the fire passes on, and no one knows where it ends.
>
> ZHUANGZI

I have presented, in turn, novel techniques for parsing into λ-calculus expressions, generating out of λ-calculus expressions, and translating between natural language sentences using λ-calculus expressions as a semantic interlingua. While these results by no means constitute a practical new machine translation paradigm on their own, they are promising: They indicate pivoting through a rich semantic interlingua can, not just in principle but in practice, produce translations of a higher quality than the standard baseline for phrase-based systems.

The fact that the GEOQUERY dataset is rarely used as a benchmark for serious machine translation systems makes it somewhat difficult to determine whether the results presented here, even for the best-performing semantic translation system, are actually an improvement over state-of-the-art phrase-based or hierarchical machine translation; I suspect they are not. But even approximate parity with the best-performing current systems would be a victory: We do not expect a brand-new model to be competitive with translation techniques that have been carefully tuned and corrected for nearly a decade. Instead these results suggest that this model will eventually outperform the current state-of-the-art systems, or at least find uses on problems (such as those requiring reasoning) where the state of the art will never suffice.

Certainly there are advantages to using this model even if it only ever performs comparably to the state of the art: Multilingual systems, for one, will benefit from having a pivot of greater richness than the pivot languages typically used today. In the past, ascending the Vauquois triangle has led not only to gains in raw translation quality, but also to a deeper understanding of the translation process and at some level, of the relationship between language and meaning. I hope that the results presented in this thesis are a step in that direction.

Bibliography

[AZ11] Yoav Artzi and Luke Zettlemoyer. Bootstrapping semantic parsers from conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 421–432, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[BCP+90] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85, June 1990.

[BCS+04] Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[BKK+77] D. Bobrow, R. Kaplan, M. Kay, D. Norman, H. Thompson, and T. Winograd. Gus, a frame-driven dialog system. *Artif. Intell.*, 8(2):155–173, April 1977.

[Blo00] Hans U. Block. Example-based incremental synchronous interpretation. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 411–417. Springer-Verlag, Berlin, 2000.

[CBKM+10] Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings*

of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pages 17–53, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[CBKMZ11] Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

[CBOK06] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of bleu in machine translation research. In *In EACL*, pages 249–256, 2006.

[Chi05] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Chu32] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2):346–366, 1932.

[dBB00] Wm. Theodore de Bary and Irene Bloom, editors. *Sources of Chinese Tradition*, volume 1. Columbia University Press, New York, New York, 2000.

[DHL04] Bonnie Dorr, Eduard Hovy, and Lori Levin. Machine translation: Interlingual methods. In Keith Brown, editor, *Encyclopedia of Language and Linguistics 2nd edition*. Elsevier, Ltd., 2004.

[GM06] Ruifang Ge and Raymond J. Mooney. Discriminative reranking for semantic parsing. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *ACL*. The Association for Computer Linguistics, 2006.

[HBG+11] Sanjika Hewavitharana, Nguyen Bach, Qin Gao, Vamshi Ambati, and Stephan Vogel. Cmu haitian creole-english translation system for wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 386–392, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

[Hue75] Gérard Huet. A unification algorithm for typed lambda-calculus. *Theoretical Computer Science*, 1(1):27–57, 1975.

[Hut97] John Hutchins. Milestones in machine translation: episodes from the history of computers and translation. *Language Today*, pages 22–23, 1997.

[JT06] Christine Jourdan and Kevin Tuite. *Language, Culture, and Society: Key Topics in Linguistic Anthropology.* Cambridge University Press, Cambridge, England, 2006.

[Kat07] Rohit Jaivant Kate. *Learning for semantic parsing with kernels under various forms of supervision.* PhD thesis, University of Texas at Austin, Austin, TX, USA, 2007. AAI3277541.

[Kay75] Martin Kay. Syntactic processing and functional sentence perspective. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, TINLAP '75, pages 12–15, Stroudsburg, PA, USA, 1975. Association for Computational Linguistics.

[KM06] Rohit J. Kate and Raymond J. Mooney. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 913–920, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[Koe04] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395, 2004.

[KZGS10] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1223–1233, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[KZGS11] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1512–1523, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[LE09] Zhifei Li and Jason Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 40–51, Singapore, August 2009.

[LJK11] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, 2011.

[LN11] Wei Lu and Hwee Tou Ng. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1611–1622, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

[LNLZ08] Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. A generative model for parsing natural language to meaning representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 783–792, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[MN94] Teruko Mitamura and Eric Nyberg. Controlled english for knowledge-based mt: Experience with the kant system. In *Proceedings of The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, 1994.

[MNC91] Teruko Mitamura, Eric Nyberg, and Jaime Carbonell. An efficient interlingua translation system for multi-lingual document production. In *Proceedings of Machine Translation Summit III*, pages 2–4, 1991.

[Och03] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[PD09] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 1–10, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[RHK06] David Reitter, Julia Hockenmaier, and Frank Keller. Priming effects in combinatory categorial grammar. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 308–316, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[SDVB01] Jean Senellart, Péter Dienes, Tamás Váradi, and Cimetière Bp. New Generation Systran Translation System. In *MT Summit VIII*, pages 18–22, 2001.

[Ste96] Mark Steedman. The Role of Prosody and Semantics in the Acquisition of Syntax. In James Morgan and Katherine Demuth, editors, *Signal to Syntax*, pages 331–342. Erlbaum, Hillsdale, NJ, 1996.

[Ste00] Mark Steedman. *The syntactic process.* MIT Press, Cambridge, MA, USA, 2000.

[TSM03] Joseph Turian, Luke Shen, and I. Dan Melamed. Evaluation of machine translation and its evaluation. In *In Proceedings of MT Summit IX*, pages 386–393, 2003.

[TSRD06] Ross Turner, Somayajulu Sripada, Ehud Reiter, and Ian P. Davy. Generating spatio-temporal descriptions in pollen forecasts. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, EACL '06, pages 163–166, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[Vau68] Bernard Vauquois. A survey of formal grammars and algorithms for recognition and translation in machine translation. In *FIP Congress-68*, pages 254–260, 1968.

[VSW94] K. Vijay-Shanker and D. J. Weir. The equivalence of four extensions of context-free grammars. *Math. Syst. Theory*, 27(6):511–546, November 1994.

[Wan80] Juen-tin Wang. On computational sentence generation from logical form. In *Proceedings of the 8th conference on Computational linguistics*, COLING '80, pages 405–411, Stroudsburg, PA, USA, 1980. Association for Computational Linguistics.

[WB03] Mike White and Jason Baldridge. Adapting chart realization to CCG. In *Proceedings of 9th European Workshop on Natural Language Generation*, Budapest, Hungary, 2003.

[Wea49] Warren Weaver. Translation. 1949.

[WM06] Yuk Wah Wong and Raymond J. Mooney. 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, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[WM07a] Yuk Wah Wong and Raymond Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[WM07b] Yuk Wah Wong and Raymond J. Mooney. Generation by inverting a semantic parser that uses statistical machine translation. In *in NAACLHLT 2007*, pages 172–179, 2007.

[Wu97] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403, September 1997.

[XM04] Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[ZC09] Luke S. Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. 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 2 - Volume 2*, ACL '09, pages 976–984, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[ZM96] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2*, AAAI'96, pages 1050–1055. AAAI Press, 1996.

[ZVOP08] Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 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, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.