Incremental Prediction and Decision-making for Simultaneous Machine Translation

by

Alvin Castillo Grissom II

B.A., Hendrix College, 2006

M.S., Emory University, 2009

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2017

This thesis entitled:
Incremental Prediction and Decision-making for Simultaneous Machine Translation
written by Alvin Castillo Grissom II
has been approved for the Department of Computer Science

Prof. Jordan Boyd-Graber

Prof. James H. Martin

Prof. Martha Palmer

Prof. Mans Hulden

Date

Grissom II, Alvin Castillo (Ph.D., Computer Science)

Incremental Prediction and Decision-making for Simultaneous Machine Translation

Thesis directed by Prof. Jordan Boyd-Graber

Recently, approaches to simultaneous machine translation—translating sentences incrementally, before they are complete—have attempted to incorporate machine learning to achieve simultaneous machine translation from verb-final languages (such as German and Japanese) to verb-medial languages (such as English). Due to the divergent syntax of these languages, particularly the head-finality of verb-final languages, simultaneous translation is a great challenge for both humans and machines, requiring the incorporation of predictions about the end of the sentences to achieve natural-sounding translations without waiting for them to be uttered. This problem is not trivial, yet relatively little attention has been given to it in the computational linguistics literature. We use incremental verb prediction to tackle this problem. By predicting verbs in SOV languages before they have been spoken, we can get ahead of the speaker, and by learning when to trust these predictions, we can minimize the propagation of error from incorrect predictions to the incremental translations. For the task of determining when to trust incrementally-revealed predictions and what actions to take based in part on these predictions, we turn to reinforcement learning. By combining incremental linguistic prediction with reinforcement learning, the simultaneous translation system can minimize translation errors introduced by relying on imperfect predictions, leading to less error-prone translations that still manage to be expeditious under realistic constraints on time.

Dedication

To my parents, Alvin and Jackie Grissom, who have supported me in ways innumerable, not only throughout an academic journey which at times seemed interminable, but from birth, modeling perseverance and never permitting me to believe that I was anything other than capable, from childhood until now.

To my grandmother, Ethel Grissom, who raised eleven of her own children, many grandchildren, and many great grandchildren, always supportive of their goals and always giving of herself.

To my grandfather, Allen Grissom (1922-2012), who, living in under the worst of Jim Crow, completed a master's degree, became a school principal, and passed his belief in education to his children.

To my grandfather, Joe Thomas (1924-2012), always my biggest fan, who, not even knowing his real name, chose one for himself, and, having only had the opportunity to gain an eighth grade education, started work as a janitor in a police department but retired as the chief of police.

To my grandmother, Ella Mae Thomas (1927-1989), who, only having a sixth grade education—the highest possible for girls in her small town— endured racism and discrimination every day but gave of herself for the success of her children and grandchildren, encouraging them to pursue higher education.

And to all of those American slaves, descendants of slaves, and their allies who fought, were wounded, and died so that I could have the opportunity to read and be read, and to those around the world facing similar struggles.

Acknowledgements

Of course, I thank my advisor, Jordan Boyd-Graber, for helping me to navigate to a timely completion of my doctoral studies in a topic of interest to me, as well as the other members of my world-class committee: Martha Palmer, James H. Martin, Hal Daumé III, and Mans Hulden. Likewise, this work would not have been possible without my other brilliant co-authors: He He, Naho Orita, and John Joseph Morgan.

I thank Yusuke Miyao for his generosity and helpful suggestions over several years, along with other former NII students for their friendship and discussions, including Giovanni Yoko Kristianto, Yoshinari Fujinuma, Han Dan, Pascual Martínez, and Sho Hoshino. I extend the same thanks to my other friends, colleagues, and collaborators from the Universities of Maryland and Colorado, including Shota Momma, Xiaolei Huang, Colin Phillips, Ellen Lau, Mohit Iyyer, Viet-An Nguyen, Forough Poursabzi, Shudong Hao, Davis Yoshida, Thang Nguyen, Naomi Feldman, Pedro Rodriguez, Fenfei Guo, Ke Zhai, and Yuening Hu. I also express my appreciation for the assistance from, and discussions with, Graham Neubig and Yusuke Oda. Additional thanks go to Daisuke Kawahara, Hiroshi Kanayama, Yuta Tsuboi, Taro Watanabe, and their groups for their discussions, in addition to Long Duong for the hundreds of conversations we had over dinner.

I extend a special acknowledgment to Shota Momma, Naho Orita, Yoshinari Fujinuma, Hiroshi Kanayama, Emiko Inoda, Yoshinori Kabeya, and Sho Hoshino for tolerating and obliging my incessant queries regarding esoteric properties of Japanese language.

I would be remiss if I did not acknowledge the earlier professors who supported me and gave me a chance, including but certainly not limited to Gabriel J. Ferrer, W. Dwayne Collins, and Eugene Agichtein.

Finally, I would like to acknowledge all of the teachers, family, friends, professors, and colleagues not named here who have contributed in ways great and small to my academic development.

Contents

Tables

Figures

Figure

Chapter 1

Introduction

稲妻やきのふは東けふは西
inazuma ya kinō wa higashi kyō wa nishi
The lightning—
Yesterday in the east,
Today the west. (Mukai Kyorai, 1651–1704)

We view the world through the lens of time, and a temporal world is one of change—one in which we know more in the future than we did in the past. Human beings, having evolved to exist in a world of temporal change, possess brains which process and navigate this world of change. We process information incrementally, and this is reflected in how we interact with the world, in the representations we use to describe it, and in how we process those representations.

1.1 On Translation

Translation is the attempt to express in one language what was originally expressed in another, and the appropriateness of a translation, more so than an abstract sense of *correctness*, depends on the context and needs of the consumer—or, in a dialogue or other two-way exchange, the needs of both the receiver and the producer of the language.

Translation theorists have long accepted that truly literal translation can be impossible (Jakobson, 1959). Consider the *haiku* at the beginning of this chapter. In my translation, I have inferred a missing preposition in the second phrase but not the third, and I made the conscious decision not to include a missing but implied verb, and to preserve the word order of the Japanese original. I did this so that the progression of the scene would not be lost. My purpose is to preserve the evocative, ephemeral nature of the moment described, in order to incrementally build the representation in a way that is as close to what the author intended as possible, as I interpret it. Were I translating a technical document, my purpose would naturally be to preserve the meaning as faithfully as possible, so that a specialist could make use of it. We change our evaluation criteria depending on our purposes.

Venuti (2008) notes that, traditionally, a (human) translation is considered, often uncritically, to be of high quality when the translator is "invisible"—that is, when the translation reads fluently and is not readily identifiable as a translation—but this a subject of debate, and simultaneous interpretation introduces a new set of complexities to evaluation (Le Féal, 1990; Lambert, 1991; Pöchhacker, 2001). And regardless of one's position on what constitutes *quality* in translation, we must *have* a position for any given translation task. Chesterman (2016) argues that "to translate without a theory is to translate blind", evoking a parallel idea pertaining to science in general:

> There is no such thing as philosophy-free science; there is only science whose philosophical baggage is taken on board without examination.
>
> (Dennett, 1995)

The most widely adhered to practice of the statistical machine translation community since Papineni et al. (2002a) has been to optimize to automatic metrics. This has drastically lowered costs and allowed us to develop systems that can automatically learn. This is the tradition that this research follows.

1.2 Incremental Language Processing

Humans process language incrementally: we create language incrementally—whether via writing, speaking, or sign language—and we likewise comprehend language incrementally. Many of our natural language processing algorithms, however, do not reflect this fact. Our tools for syntactic analysis typically require an entire sentence to work reliably; our classification problems assume the availability of a static set of features with which to train a model; and most machine translation research focuses on the easier (but still extremely difficult) problem of translation of static sentences.

Machine translation—algorithmic translation from one language to another—has largely exemplified this limitation. Traditionally, most machine translation research has focused on the problem of translating entire sentences at once. This is the case where we have an entire sentence in a source language and we would like to translate it into some target language. This is a case of perfect information: the translator has access to all available resources, and there is, in principle, no limit to the time to be spent on analysis. Contrast this to the task of the simultaneous interpreter, human or machine. The interpreter must, incrementally, understand what is being said in a source language, translate it in real time into the target language, and do so in a way that maximizes fluency when there is missing information. This is a much more difficult problem.

This problem, additionally, forces us to make some decisions. Not only must we decide how to evaluate such translations, we must decide how to produce them. Statistical machine translation systems typically translate a sentence at a time, without reference to the overall context. Of course no professional translator would do this. There would be copious editing and backtracking as the translation process continued and the translator gained a better understanding of the material. A single phrase or even word may be pored over indefinitely. Our current machine translation systems do nothing even approaching this.

In speech-to-speech simultaneous interpretation, without literal time traveling, there will be no backtracking. There may be *post-hoc* corrections, but we cannot edit what has been spoken. In speech-to-text or text-to-text simultaneous translation, however, we have the theoretical option of editing. A recent speech-to-text translation system uses this approach. (Müller et al., 2016).

Consider, as well, how difficult this problem is when these languages are distant structurally—for example, if the word order is reversed or otherwise divergent. Translating between distant language pairs is already inherently more difficult, for both humans and algorithms, than translating between closely related languages. Doing so incrementally, and under time and information constraints, is even more so.

This is a fundamental problem. In German and Japanese, for example, it is common for sentences to have subject-object-verb (SOV) word order—in Japanese, it is almost always the case—whereas in modern English, subject-verb-object (SVO) is canonical. If the main verb appears at the end of a German sentence, if we would like to avoid waiting until the end to begin interpreting, we must incorporate some strategy for doing so fluently. The one pursued in this research is that of *verb prediction*.

By predicting the main verb in a source language sentence in which the verb comes at the end and applying it earlier in the translation, we can start translating before the verb has been observed directly, reducing the delay of the translation. Naturally, no prediction will be correct all of the time, and, as we shall see, verb prediction is an extremely difficult problem (Chapter 3). We need, therefore, some way of deciding when to trust these predictions. For this we turn to reinforcement learning (Chapter 4), which allows us to learn to make decisions about when to trust these predictions. This work represents only the beginning of reinforcement learning-based predictive approaches to simultaneous machine translation.. We predict only verbs, for example, while there are many other properties of sentences we could predict. But we here lay a foundation for this way of attacking simultaneous machine translation and similar problems.

Historically, there has been much debate and consternation on the relative merits of statistical versus hand-written, rule-based approaches to computational linguistics (Chapter 2). In this research, we rely on statistical machine learning for both verb prediction and decision-making. But while this method relies heavily on machine learning, linguistic analysis is used to determine those features from which the machine learning algorithms can learn more effectively (Chapter 3). In addition, we shall see that, even when our aim is practical application—in this case, simultaneous interpretation—we can glean linguistic insight from our experiments that increase our understanding of human language processing. This, in turn, has the potential to inform our future approaches to mimicking these processes.

We begin by introducing the requisite computational background for machine translation and simultaneous machine translation in Chapter 2. We then outline our discriminative approach to the prediction of final verbs in Chapter 3. Chapter 4 presents an idealized version of our system that abstracts away the translation problem from the machine learning problem. We introduce the notion of an *omniscient translator*—an oracle translation system that enables us to focus on the learning problem by making translation more predictable than a production translation system. Finally, having demonstrated that learning can take place in a predictable setting, in Chapter 5, we describe a more realistic system: we incorporate improved verb prediction models, described in Chapter 3, into the system, as well as a phrase-based translation system and a more robust language model for predicting upcoming words. We show that, under these circumstances, the system can learn, just as in the ideal case.

Chapter 2

Computational Background

We begin by reviewing some preliminary background information on computational approaches to machine translation, including the evolution of machine translation research, earlier approaches to simultaneous machine translation, and automatic evaluation metrics for machine translation systems. We also review some preliminary information on reinforcement learning, which is key to research described here.

The problem domain of incremental classification has recently been approached from the standpoint of question answering in the computational linguistics literature literature. (Boyd-Graber et al., 2012; Iyyer et al., 2014) This is the problem setting in which an algorithm must make decisions, not only under uncertainty, but under time constraints. In such settings, given available information at a given point in time, the algorithm must decide whether or not to trust the information to which it has access. One application of this is simultaneous machine translation.

The problem of linguistic prediction has recently attracted attention in the computational linguistics literature. This is the problem whereby an algorithm must make decisions, not only under uncertainty, but under time constraints. In this setting, given available information at a given point in time, the algorithm must decide whether or not to trust the information to which it has access. A recent application of this is simultaneous machine translation.

Simultaneous translation in languages with divergent canonical word order contrasts with simultaneous translation of other languages. In this research we consider simultaneous translation from languages with frequent subject-object-verb (SOV) word order to languages with subject-verb-object (SOV) word order; in particular, we consider German-English translation, and we consider prediction of the sentence-final verb in both German and Japanese.

In Japanese, a strongly verb-final language, the main verb appears at the end of a clause, almost invariably. For Japanese-English simultaneous machine translation, the first (and, to our knowledge, only) prior work (Matsubara et al., 2000b) to use verb prediction used pattern matching-based verb predictions *before* the input sentence is complete in order to outpace of the speaker in translation. Theirs was the first and only attempt, until the present work, to use prediction for simultaneous machine translation. Their use of verb prediction relied on basic pattern matching and was not linguistically informed, but there is evidence suggesting that linguistically-motivated features would prove useful in making this problem tractable (Chapter 3). The research in this dissertation frames simultaneous machine translation as an instance of incremental classification by specifically using reinforcement learning to decide when to act and what the action should be. This follows the similar approach (Boyd-Graber et al., 2012), which used reinforcement learning to answer incrementally revealed questions in a trivia game, Quiz Bowl. In this section, we review some of the relevant literature for incremental classification and computational approaches to linguistic prediction.

2.1 Machine Translation

2.1.1 Beginnings

Machine translation (MT) is the computational process of translating from one language to another. This was first attempted in the 1954 Georgetown-IBM experiments (Dostert, 1955), the goal being to translate Russian into English. At the time, machine translation was viewed as a deterministic decision problem (Garvin, 1968) that could be solved in five years. These systems were syntax-based (that is, rule-based): human beings manually created rules to map between the two languages. While the researchers thought this to be straightforward, in this case of over-promising and under-delivering, machine translation research suffered for the next thirty years—that is, until IBM revolutionized the field with its Bayesian approach (Brown et al., 1990), which applied Bayesian inference to the problem of creating a translation model, an approach that remains the dominant paradigm to this day. Prior to this, however, there had been corpora-based work on

Japanese-English machine translation (Nagao, 1984; Sumita et al., 1990)

The *Mu* project was an early approach that used what is called **example-based** translation (Nagao, 1984). This empirical approach infers rules by finding parallel phrases and replacing a single word. This allows the system to infer rules that can be applied to other, unseen examples. For example, if, given the pairs (*furansu no ō*, King of France) and (*furansu no tabemono*, food of France), the system may infer the rule that maps "X *no* Y" to "Y of X", where X and Y are chosen based on similarity in a constructed thesaurus. Such transfer rules can then be combined hierarchically to construct a translation. Ambiguities are resolved by measuring the semantic distance between candidates, theoretically obtained from a thesaurus which contains such measures. This is motivated explicitly by observations of human learning of translation patterns and is an instance of case-based reasoning (Schank, 1983). They claim the following the following:

> (1) Man does not translate a simple sentence by doing deep linguistic analysis, rather,
>
> (2) Man does the translation, first, by properly decomposing an input sentence into certain fragmental phrases (very often, into case frame units), then, by translating these fragmental phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference. . .

(Nagao, 1984)

This argues for learning mappings from corpora to translate between Japanese and English: it is a rule-based approach, but an approach which extracts rules from the corpus. It is not statistical.

Since the statistical shift in natural language processing took place (Brown et al., 1990), probabilistic approaches have dominated machine translation research since they have been largely effective for the languages under study. With the advent of the BLEU (Papineni et al., 2002b) metric for machine translation quality, moreover, translation systems have a clear objective function to optimize, without need of expensive human judgments in the process. A notable consequence of this, however, is that, since the BLEU metric often does not work well with English and Japanese (Section 2.5.2), despite the early pioneering work on Japanese machine translation, the most prominent metrics being optimized are not appropriate for translation between Japanese and the most commonly studied European languages.

Few dispute the notion that statistical machine translation advanced (perhaps rescued) the field. But simultaneous machine translation, which has historically received less attention, still relied on hand-crafted rules long after machine translation itself had largely made the shift to statistical approaches.

2.2 Phrase-based Machine Translation

Today, statistical phrase-based approaches to machine translation are more popular, and we employ phrase-based machine translation in our research (Chapater 5). Fundamentally, phrase-based approaches to machine translation all build on the innovations of the IBM system. (Brown et al., 1990). Such systems require a parallel corpus—a **bitext** (Harris, 1988)[1] —that contains the same text (usually sentences) in both the source and target language. It is then the job of algorithms to automatically infer the relationships between them to ultimately generalize to unseen texts.

The first step in such a system is to create word and phrase **alignments**. This is the probabilistic process of determining, from the bitext, which chunks of text in the source language align with other chunks of text in the target language. While there are several methods of accomplishing this, the most common are hidden Markov model (HMM)-based methods (Vogel et al., 1996a; Toutanova et al., 2002; Deng and Byrne, 2008) and Bayesian IBM Model n-based methods (Brown et al., 1993) available in widely-used software such as GIZA++ (Och and Ney, 2000).

The IBM Model 1 algorithm, on which the subsequent IBM models are based, creates a mapping from a source word f_k to a target word e_j. Some words have no alignment; others can be added. These alignments are calculated by maximizing the probability that they occur together. That is, given a foreign sentence $\mathbf{f} = (f_1, ..., f_n)$, native sentence $\mathbf{e} = (e_1, ...e_m)$ and an alignment $\mathbf{a}$, by Bayes's Rule, we maximize

$$P(\mathbf{e}, \mathbf{a}|\mathbf{f}) = P(\mathbf{e})P(\mathbf{f}|\mathbf{e}). \tag{2.1}$$

[1] The notion of a "bi-text" described by Harris (1988) is different from the one used by the statistical translation community today.

> One way to describe bi-text, therefore — and this is a basic definition — is to say that it is [source text] and [target text] as they co-exist in the translator's mind at the moment of translating. At that point [the target text], still in the creation and growing process, has not yet gone its own way; but even later on, those of us who are bilingual and who want to study the translation can also 're-store' both versions, or at least parts of them, in our minds simultaneously and consider them together. It is a translation in this state, when it is not a separate [srouce text] and [translation text] but the two sewn firmly together like a piece of cloth and its lining to be used as one fabric, which constitutes a bi-text for others as well as for the translator. (Harris, 1988)

His focus lies more so in tools for professional translators than in building an automatic translation system.

The approach, called the **noisy channel model**, has its roots in a memorandum (Weaver, 1955), in which Weaver described translation in cryptographic terms—drawing upon recent innovations in information theory by Shannon (1948)—who explicitly used the terminology of the "noisy channel"— describing translation as "decoding", a term which would be picked up again by Brown et al. (1990) and become a fixture of statistical machine translation vernacular with the proliferation of this approach.

> We now consider the case where the signal is perturbed by noise during transmission or at one or the other of the terminals. This means that the received signal is not necessarily the same as that sent out by the transmitter. Two cases may be distinguished. If a particular transmitted signal always produces the same received signal, i.e. the received signal is a definite function of the transmitted signal, then the effect may be called distortion. If this function has an inverse—no two transmitted signals producing the same received signal—distortion may be corrected, at least in principle, by merely performing the inverse functional operation on the received signal. (Shannon, 1948)

We use Bayes's Rule (Equation 2.1) to create a generative model, and frame it as a noisy channel. Variations of this model are used not only in machine translation, but spelling correction and speech recognition, among other domains.

A **phrase** in machine translation refers to a contiguous sequence of words. In phrase-based machine translation, we attempt to map phrases from the source language to those in the target language. The simplest way of accomplishing this is to use the relative frequency of their co-occurrence. These may then be stored in a phrase table to estimate the probability that one phrase aligns to another. In the machine translation case, we tell the following story: We have some text (say, German) which we suppose was *generated* by English, but it was transmitted to us through a "noisy channel" (e.g., encrypted or distorted, depending on the analogy). The corrupted English that we receive, then, is the German text we see, and we must decode it back into its original English. We thus find the alignment with the highest probability of Equation 2.1. Our prior, $P(\mathbf{e})$, is the probability of the English text, and $P(\mathbf{f}|\mathbf{e})$ is the probability that the English translation (decoded) gave rise to the German we received. The prior term, $P(\mathbf{e})$ is typically calculated with an n-gram language model.[2]

[2] Solomonoff (1957) was the first to describe n-gram language models.

> An n-gram is a sequence of objects – in our case, words – but the "words" could be bits, letters, or anything else, in principle. Given the sentence, I was late, (I), (was), and (late) are unigrams, (I was) and (was late) are bigrams, and (I was late) is a trigram. Beyond level three, usually, these are simply referred to as 4-grams,

Other phrase-based translation systems construct a **grammar** that determines how a phrase can be mapped between languages. This may be accomplished with a probabilistic context free grammar (PCFG) or other formalisms. Hierarchical phrase-based translation (Chiang, 2007) extends the notion of a grammar to include gaps: in such systems, the phrases need not be contiguous. This kind of grammar is obviously more complex, but also allows for greater expressibility.

2.3 Simultaneous Machine Translation

Simultaneous machine translation, the machine-based equivalent to simultaneous interpretation, is the algorithmic process of translating sentences incrementally: rather than assume that the source language sentence is complete, the system constructs the translation as best it can as new content is observed. In the case of SOV to SVO translation, since the main verb may appear later in an SOV language, for the translation to be truly incremental, the verb, copula, or other sentence-final components must be predicted and translated before they are actualized in the source sentence. Despite its importance, this topic is as yet mostly unexplored.

2.3.1 Example-based SOV-SVO Simultaneous Machine Translation

Furuse and Iida (1996) describe an incrementalized version of the example-based transfer grammar induction proposed by Nagao (1984) for Japanese to English. This is extended by Mima et al. (1998a), who apply same approach (constituent boundary patterns) to simultaneous machine translation from Japanese to English, noting from the start that defining appropriate information unit—atoms adequate for incremental

5-grams, etc. (Solomonoff, 1957)

But Solomonoff (1957) was probably influenced by Weaver (1955).

> First, let us think of a way in which the problem of multiple meaning can, in principle at least, be solved . If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. “Fast” may mean “rapid”; or it may mean “motionless”; and there is no way of telling which. But, if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. side, then, if N is large enough one can unambiguously decide the meaning of the central word. The formal truth of this statement becomes clear when one mentions that the middle word of a whole article or a whole book is unambiguous if one has read the whole article or book, providing of course that the article or book is sufficiently well written to communicate at all. (Weaver, 1955)

translation in both the source and target languages—is crucial. They aim to explicitly incorporate the knowledge of a simultaneous interpreter into their system. They use the following as the kinds of sentence-planning examples that would be useful for such a system:

- **Tag Question Correction** Japanese tag questions (e.g.., "isn't it?" or "right?" in English) indicated at the end of sentences. To handle this, when the final question particle is observed, the system appends it to the English translation (i.e., "isn't it?")
- **Negation Correction** Save for sentence-final particles (including tag questions), negation is the last morpheme to occur in Japanese clauses and sentences. They correct for a specific case of negation: negation followed by a contrastive conjunction ("but").
- **Prediction Failure Correction** They list a specific example of using an incorrect prediction, but grammatically making the final sentence sensical.

To implement these strategies, they augment their example database with data from simultaneous interpreters, and, in addition to the usual semantic similarity criteria used in example-based translation, check for the contextual consistency between the next phrase and the previous phrase. This, the key contribution, is accomplished via their method of example selection: namely, selecting examples with previous context similar to that of the new translation.

The issue of word order in Japanese to English simultaneous machine translation was further explored by Matsubara et al. (2000b), who tackle the quandary of optimizing competing objectives: that of immediacy (how fast the translations are) and quality (how good they are). Their techniques are fivefold: prediction of the English verb in advance, phrase reordering, changing the voice (e.g., passivization), changing the intonation (in a speech context), and generating repairing expressions.

2.3.2 Learning Optimal Segmentation Strategies

In simultaneous machine translation, deciding *when* to translate is paramount. The naïve approach of waiting until all input is observed is not simultaneous translation; thus, determining the appropriate segments for translation is a necessary step in creating such a system in an SOV-SVO context.

The Verbmobil project (Wahlster, 2000) also uses syntax-based approaches to achieve simultaneous translation between German or Japanese into English. To produce grammatical sentences in English, this system incorporated natural language generation on the English side to ensure syntactically well-formed target expressions. Verbmobil also maintains a domain model of relevant semantic information. The system does not explicitly account for the SOV-SVO problem.

Fujita et al. (2013a) uses the probabilities in a phrase table learned from corpora to optimize Japanese to English incremental segmentation. The strategy is the following: as each Japanese word is observed, the system checks for its existence as part of a phrase in the phrase table. If it matches a phrase pattern, the system waits for more words. Otherwise, the system outputs the cache of all previously untranslated words. The idea is to translate entire phrases, rather than word units that are too small to be translated accurately as atoms. A cache is built to match the longest phrase that exists in the phrase table; once adding a new words creates a phrase with no matches in the phrase table, the cache is output. To translate across phrase boundaries, they use the *right probability* of each phrase, which they describe as follows:

> [T]he RP is formally defined as the probability that when the current phrase ends at words fj and ei in the source and target respectively, the source phrase be- ginning at f_{j+1} is aligned to a target phrase starting at e_{i+1} or later. In other words, The RP indicates the probability that the phrases occur in the same order in both languages, and thus directly correlates to our goal of judging when we can segment the input sentence without disturbing translation order.

They use this probability to determine whether to translate the phrase hits in the phrase table. They heuristically set the threshold manually, to favor either waiting or translating more quickly. If the right probability of a given phrase is lower than the threshold, it is not output; otherwise, the entire cache is output.

Yoshikawa et al. (2005) segment at the clause-level, using the Japanese clause segmenter CBAP (Maruyama et al., 2004; Ohno et al., 2005). Intuitively, clauses map reasonably well between Japanese and English. They use the following example to illustrate the similar clause boundaries:

- ホテルの予約をして来なかったので /

  こちらでホテルを紹介していただきたいんですけれども

  hotel-GEN reservation-ACC come-not-PAST **because** /

here hotel-ACC refer want but

- I haven't made any hotel reservations /

  so, could you please introduce me to any nice[sic] hotel?

The approach is intuitive, but they find that most of the sentences would only be segmented into one unit using this method, and the experiments were limited to clauses which appear in the same order in both Japanese and English.

Oda et al. (2014) takes a machine learning approach to optimizing segmentation, with the goal of maximizing an automatic evaluation measure: in their case, BLEU and RIBES (Section 2.5.2). They learn a segmentation model that segments based on groups of features. Each potential phrase boundary has tokens and features associated with said tokens; these are used to learn which features indicate an appropriate segmentation boundary. For features, they use the part-of-speech tags surrounding each potential boundary. Since the number of potential boundaries is exponential in the length of the sentence, they use dynamic programming to compute these. Additionally, they introduce the constraint that all segmentation positions with the same part-of-speech tags must be selected at the same time. The algorithm finds all features in the corpus that occur exactly j times and measures the effect of adding them to the best segmentation strategy at that iteration. When compared to a greedy search baseline, this significantly outperforms using right probability (Fujita et al., 2013a) when optimizing to both the precision-oriented RIBES and the reordering-sensitive RIBES scores. We use machine learning—specifically, reinforcement learning—for learning translation strategies in an *online* setting (Chapter 4).

2.4 Reinforcement Learning

Reinforcement learning (Kaelbling et al., 1996) algorithms are a class of machine learning algorithms broadly inspired by behaviorist psychology. Typically, they are used for learning strategies for navigating a complex environment. More specifically, reinforcement learning systems learn a **policy** to navigate a stochastic search space. The search space is usually formulated as a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998). In a POMDP, an agent lacks complete access to the state space, instead estimating the probabilities of the state space based on its observations. Given these estimates, the agent takes an **action**, moving it to a new state. Unlike in HMMs, the selected transition from one state to the next is not only a function of the probabilities of the transitions (which may be unknown), but determined by the action taken in a given state. Reinforcement learning is a formulation for optimizing this class of PSPACE-complete(Papadimitriou and Tsitsiklis, 1987) problems. In general, we have:

- a finite set of states, S
- a finite set of actions, $A \times S \to S$ that transition to a new state.
- a reward function $R : S \times A \to \mathbf{R}$ that determines the immediate reward of taking an action a in state s.
- Ω a set of observations in state s
- a **policy** $\pi : \Omega \to A$ that maps states to actions. I.e., if we are in state s_t, the policy decides which action to take, moving us to state s_{t+1}.

At any given point, the agent is in a state $s_t \in S$, but it may be intractable to fully observe the state space. We want to learn a policy, π, that maps states to actions in such a way as to maximize the final reward, even given incomplete knowledge of the state space. That is, given what can be observed in s_t, the policy π selects an action to move the agent to state s_{t+1}. Each action selection incurs a **reward** (or, inversely, a **cost**) based on the properties observable in s_{t+1}. This reward modifies the parameters of π. This process continues until a termination state is reached.

In a reinforcement learning-based simultaneous translation system (Chapter 4), we do not know what words will come next; we only know the words that have been thus far observed, our predictions of the next word and the final verb, and the probabilities of those predictions. Based on this information, we must to decide which action to take. This system has four possible actions: do we **wait** for more input, trust our **next word** prediction and use it, trust our prediction of the final **verb** and append it, or merely **commit** to translate the cache of words we have yet to emit? Given these possible actions, the **policy** must decide which one to take with each new observation.[3]

2.4.1 Imitation Learning

Imitation learning is a kind of reinforcement learning whereby the policy is learned from an expert, called an **oracle**. We use SEARN (Daumé et al., 2009), an imitation learning algorithm for learning from an oracle policy. The oracle policy π^* is the teacher; the learning policy π is the apprentice. In imitation learning, we want an apprentice policy π that can, after some training, imitate the policy of the oracle policy π^*. Imitation learning provides a kind of supervision for reinforcement learning in the form of the oracle.

A common problem, however, is that the oracle policy is so much better than the learning policy at maximizing the reward that the learning policy cannot learn effectively from it. Intuitively, this is because the oracle policy may not even have explored similar states to the learning policy, due to the great gulf between the decisions that they make. From very early in the decision process, the trajectory of the oracle policy may deviate substantially from that of the apprentice policy, making it impossible for the apprentice to learn what to do in that states in which it finds itself. In much the same way as a first grader will likely not learn much from a class in vector calculus, a policy that is just starting to learn will likely not learn effectively from a policy that has little to no overlap in the kinds of situations in which the apprentice policy finds itself.

2.4.2 Imitation Learning as Classification

Imitation learning is a kind of reinforcement learning whereby we have a policy, π, can be reduced to classification. (Syed and Schapire, 2010) In this formulation, the agent is a classifier wherein the state, s_t, is represented as a feature vector. The classifier, then, selects an action based on these features. We additionally have an oracle, π^*, which supervises π, facilitating the reduction to classification. In this case, to compute the cost, we compute the disparity between the score of the optimal action, determined by π^*, and the action selected by π, modifying the weights in the classifier in proportion to this difference.

When we learn a classifier policy, we train a multiclass classifier which, based on the state s_t, selects an

[3] While we thus far have framed statistical approaches in contrast to the “decision-problem” based approaches of rule-based system, it is worth noting that the the reinforcement-learning based formulation is fundamentally a series of decision problems—specifically, it is a Markov Decision Process.

action. Here, s_t is a vector of features extracted from the observations. More informative features will allow a classifier, π, to be learned which better estimates the actions that will lead to a higher-scoring trajectory.

Supervision, as described earlier, takes place in the form of an optimal policy, π^*, which provides training examples for the classifier. The particular method of imitation learning determines how the supervision is applied. The optimal policy need not be truly optimal; it must, however, provide examples from which the classifier can learn.

Classifiers learn effectively because the data on which they are trained can effectively generalize to unseen data. When the data on which the classifier is tested has too little in common with the data on which it is trained, effective generalization cannot occur. We call this *overfitting*. If we train π with an oracle, π^*, we run the risk of overfitting to π^*.

This is where imitation learning approaches such as SEARN become useful: instead of a clueless learner learning from a perfect learner, the learning policy learns by interpolating its decisions with those of the oracle policy. Intuitively, the first grader copies the instructor some of the time and does his own work for the remainder of the time. Earlier in the course, the first grader copies (imitates) more often; later in the course, he does more of the work on his own. More formally, we can define the policy selection as the following:

$$\pi_{k+1} = \epsilon\pi_k + (1 - \epsilon)\pi^* \tag{2.2}$$

In words, on iteration $k + 1$, the action selected comes from the oracle policy with probability $1 - \epsilon$ and the k-th iteration's apprentice policy with probability ϵ, making the respective state spaces in which they find themselves similar enough for the learning policy to learn from the oracle. The cost of each action is defined as:

$$\mathcal{C}(a_t, \boldsymbol{x}) \equiv Q(\boldsymbol{x}, \pi^*(x_t)) - Q(\boldsymbol{x}, a_t(x_t)), \tag{2.3}$$

where $a_t(x_t)$ represents the translation outcome of taking action a_t and the Q function is our objective function that evaluates the policy outcome (e.g., the translation). That is, the cost of each action is the difference between the respective scores of the optimal action that π^* would choose and the action actually chosen by the apprentice policy, π. evaluation

2.5 Evaluation of Machine Translation Systems

Given the difficulty of evaluating standard machine translation, it is a given that the evaluation of simultaneous machine translation is nontrivial. The rapid progress in machine translation technology is largely made possible by less of a need for human judgments in translation systems. This is accomplished with metrics known to correlate with human judgments. Most of the systems surveyed here were created before the widespread usage of automatic evaluation metrics. The systems of Fujita et al. (2013a) and Oda et al. (2014) rely upon current evaluation metrics (BLEU and RIBES). We create our own evaluation metric (Section 4.4) based on BLEU ((Papineni et al., 2002a)), which takes into account the delay in translation.

2.5.1 NIST

NIST (Doddington, 2002) is an automatic evaluation metric based on BLEU. It augments the n-gram matching of BLEU with *information weights*, using n-gram counts over the reference translations. This results in an increased correlation with human judgments.

$$\text{Info}(w_1, ..., w_n) = \log \frac{Count(w_1, ..., w_{n-1})}{Count(w_1, ..., w_n)} \tag{2.4}$$

They also modify the brevity penalty to minimize the impact of small variations in length. More relevant to the current topic, in their initial experiments measuring correlation of n-gram precision to human fluency judgments, among systems from French, Chinese, Japanese, and Spanish, Japanese was the only one which had a poor correlation. This poor correlation (for both fluency and adequacy) is maintained across BLEU and NIST.

2.5.2 RIBES

We are considering, specifically, SOV-SVO machine translation, which necessitates that we consider word order. The RIBES (Isozaki et al., 2010) metric is commonly used, often in conjunction with BLEU, for the evaluation of machine translation between Japanese and English. This is, they note, because the precision-based BLEU often gives high scores to misleading and terrible translations, simply because the

n-grams match. This is a particularly serious problem in Japanese to English translation (perhaps even more so than with other SOV-SVO language pairs) since modern English generally lacks overt case. While a case-rich language might be able to recover from order inconsistencies, when the word order strongly determines the syntax and semantics of a sentence, as in English, this is a fundamental problem. BLEU also assumes the existence of word boundaries, but it is often unclear what counts as a word boundary in Japanese (Denoual and Lepage, 2005). Goto et al. (2013) argues, based on acceptability comparisons between NIST, BLEU, and RIBES that NIST and BLEU are inappropriate for use with Japanese and English. In this meta-evaluation, RIBES outperforms both NIST and BLEU for both J-E and E-J translation.

RIBES, not without its own problems, attempts to address the word order problem. RIBES is based on rank coefficients. Isozaki et al. (2010) describes results using Pearson's ρ and Kendall's τ. Here we focus on the version that uses Kendall's τ, but the principle is the same for both.[4] Since these metrics only work when there is a one-to-one correspondence between lists—an assumption that does not hold when considering token counts between languages—they instead only consider one-to-one corresponding bigrams and remove non-aligned words entirely from the calculation.

$$\tau = 2\frac{\text{number of increasing pairs}}{\text{number of all pairs}} - 1, \tag{2.5}$$

where the *increasing pairs* refers to the number of words in the machine translation whose indexes are increasing as in the reference translation.

Consider the following example from Isozaki et al. (2010):

- R: John$_1$ hit$_2$ Bob$_3$ yesterday$_4$
- M: Bob$_3$ hit$_2$ John$_1$ yesterday$_4$.

The machine's translation order is now [3, 2, 1, 4]. This has three increasing pairs: $(3, 4)$, $(2, 4)$, and $(1, 4)$. Thus, we have $\tau = 0$.

They use a normalized Kendall's τ (NKT) to project onto $[0, 1]$:

$$NKT = (\tau + 1)/2 \tag{2.6}$$

[4] The implementation available for download from NTT uses Kendall's τ

They show that, for Japanese and English, their metric performs similarly to other metrics in meta-evaluation, but significantly outperforms them for distant language pairs. They found that using a brevity penalty does *not* correlate well with human judgments of adequacy and is even counter-productive. They found that precision P better correlated with human adequacy judgments than the brevity penalty.

$$P = \frac{c}{\|h\|}, \tag{2.7}$$

where c is the number of corresponding words and $\|h\|$ the number of words in the hypothesis sentence h.

Multiplying NKT by P results in too much of a penalty; so, their final metric takes a higher order root of P and multiplies it by NKT.

$$NKT \times P^{\alpha} \tag{2.8}$$

where $[0 \leq \alpha \leq 1]$.

The nature of this metric—being based only on order—makes it potentially suitable for incrementalization.

2.5.3 Semantics-based metrics

The MEANT[5] (Lo and Wu, 2011) evaluation metric, used in standard machine translation, is uniquely appealing for application to simultaneous machine translation due to its overt focus on semantics. While RIBES implicitly focuses on semantics for certain language pairs by focusing on word order, MEANT measures the overlap of semantic role fillers. In simultaneous interpretation, the objective is not necessarily to correctly translate every nuance: the goal is to communicate the essentials—basically, who did what to whom, why, where, and how—while not falling behind the speaker. Like RIBES, MEANT focuses on **adequacy**. The premise behind the metric is that, for translation to be useful, one must at least understand the basic event structure; hence, eschewing the focus on lexical decisions, the focus is shifted to basic semantic role preservation. MEANT functions by making use of prior labeling of semantic roles in the reference translations and the machine translations. Then, a weighted F-measure is used to measure the precision and recall of

[5] MEANT is a family of algorithms. HMEANT is the version that relies on human labeling

the overlap. Lo and Wu (2011) report that using automatic labelers can achieve 80% of the performance of humans. They show that their metric correlates with human judgments far better than BLEU. The MEANT metric quantifies the following:

- $C_{i,j}$ = number of correct fillers of ARG j for PRED i in MT
- $P_{i,j}$ = number of partial fillers of ARG j for PRED i in MT
- $M_{i,j}$ = total number of fillers of ARG j for PRED i in MT
- $R_{i,j}$ total number of fillers of ARG j of PRED i in REF

Described in Lo and Wu (2011), for each of these, a weighted precision and recall is calculated in the standard way by measuring semantic role overlap. From these, overall precision is calculated:

$$\text{precision} = \frac{C_{precision} + (w_{partial} \times P_{precision})}{\text{total num. predicates in MT}} \tag{2.9}$$

$$\text{recall} = C_{recall} + (w_{partial} \times P_{recall}) \tag{2.10}$$

$$F1 = \frac{2(\text{precision})(\text{recall})}{\text{precision} + \text{recall}} \tag{2.11}$$

where w_{pred} is the weight for partial is the weight for partially matched partial translations.

Ten labels evaluated in their MEANT: AGENT (who), ACTION (did), EXPERIENCER (what), PATIENT (whom), TEMPORAL (when), LOCATION (where), PURPOSE (why), MANNER (how), DEGREE (how), and other (how).

Compared with other measures for sentence-level human adequacy judgments, by Kendall's τ, HMEANT scores 0.4324, exactly the same as HTER, which is much higher than the next highest scorer, NIST, which scores 0.2883. Next are BLEU and METEOR, which only score 0.1892. The lowest performer was word error rate, at 0.991.

2.6 Conclusion

In this section, we have surveyed the fundamental background information for our computational approach to SOV-SVO simultaneous machine translation. The present work proceeds in the spirit of statistical approach to machine translation, but we also incorporate linguistic insight into our computational system. In the following chapter, we review relevant linguistics work on verb prediction in humans and present our work on verb prediction.

Chapter 3

Prediction of Final Verbs

In Chapter 2, we introduced the fundamental computationl concepts employed in our approach to verb-final simultaneous machine translation. In this chapter, we turn our attention to the actual prediction of sentence-final verbs[1] . A key component of this research is the incremental prediction of verbs in SOV languages. We know that humans predict future linguistic input before it is observed (Kutas et al., 2011). This predictability has long been formalized in information theoretic terms (Shannon, 1948), as higher entropy indicates lower predictability. explanations of many linguistic phenomena, such as garden path amiguity. (Den and Inoue, 1997; Hale, 2001). This kind of linguistic prediction is fundamental to statistical natural language processing. Efficient auto-completion in search engines, for example, has made next-word prediction one of best known applications in natural language processing.

This kind of prediction has also been used to aid machines in the task of simultaneous machine translation. Long-distance word prediction, such as verb prediction in SOV languages (Levy and Keller, 2013; Momma et al., 2015; Chow et al., 2015; Momma, 2016), is one important strategy in simultaneous machine translation from subject-object-verb (SOV) languages to subject-verb-object (SVO) languages. In SVO languages such as English, for example, the main verb phrase usually comes after the first noun phrase—the main subject—in a sentence, while in verb-final languages, such as Japanese or German, it appears very last. Human simultaneous translators must make predictions about the unspoken final elements in a sentence (such as verbs) in order to incrementally translate it. Minimizing interpretation delay thus requires making constant predictions and deciding when to trust those predictions and commit to translating in real-time.

[1] This work originally appeared in Grissom II et al. (2016)

In Chapter 2, we reviewed previous work in predictive approaches to simultaneous maahine translation using pattern-matching rules (Matsubara et al., 2000a) and the prediction of entire syntactic constituents (Oda et al., 2015) for Japanese-English simultaneous translation These systems require fast, accurate prediction of late-occurring words to further improve simultaneous translation systems. We focus on verb prediction in verb-final languages with this motivation in mind. Section 3.1 reviews the linguistic preliminaries and relevant prior work on verb prediction. Section 3.2 presents the first systematic study of humans' ability to incrementally predict the verbs in Japanese. We use these human data as a yardstick to which to compare computational incremental verb prediction. Incorporating some of the key insights from our human study into a discriminative model—namely, the importance of case markers—Section 3.3 presents a better incremental verb classifier than existing verb prediction schemes.

3.1 Background

3.1.1 Word Order And Case Marking

We now review relevant previous work on how aspects of sentences affect sentence processing in humans. Previous work in linguistics has studied how case and word order affect sentence processing in native Japanese speakers. What is difficult or easy for humans can offer clues for what will be useful for a computational system.

Japanese is a strongly verb-final language. Nearly all sentences, save for occasional marked exceptions in speech, end in verbal words,[2] including verbs, adjectives, and copulas, all of which have inflectional morphology to denote tense and aspect, mood, and polarity. Since the verb phrase comes at the end, several theories have been posited to account for the incremental processing of Japanese sentences. If humans make successful predictions about upcoming verbs, it is instructive to examine what we do know about the information that they use to do so.

[2] Well-formed Japanese sentences may end in verbs, "adjectives", and copulas, with only idiosyncratic exceptions, though in informal language, arbitrary chunks of the sentence may be omitted and inferred by the listener or reader from context. Contrary to folk belief, this is not limited to the subject and topic. Japanese sentences may end in "adjectives" (with an implied copula following), but according to some analyses, Japanese does not have adjectives. More specifically 形容 詞 (*i*-adjectives) can be analyzed as verbs, and 形容動 詞 (*na*-adjectives) can be analyzed as verb phrases. This is an area of ongoing research and debate (Backhouse, 1984; Wetzer, 1996; Namai, 2002; Baker, 2003). We take no definitive position on the issue here.

Verbs constrain the possible structures that an incomplete sentence can eventually take. Since Japanese verbs arrive at the end of clauses, Japanese sentences have a number of competing possible structures before the verb is observed. Yamashita (1997) investigates the effect of word order and case markings on the processing of Japanese verbs. Consider their example of an incomplete sentence:

- メアリーが　ジョンに　りんごを
  Mary-NOM John-DAT apple-ACC
  "Mary [missing verb] John [an] apple."

This could be subcategorized by a matrix clause, a sentential complement, the final verb, or a mixture of verbs, as shown in their completion examples, a few of which are shown below:

- [メアリーが ジョンに [[EC りんごを (食べた] 人]を 紹介した])
  Mary-NOM John-DAT [[EC apple-ACC (eat-PAST person]-ACC introduce-PAST])
  "Mary introduced the person who ate an apple to John."

- [メアリーが ジョンに りんごを [[EC (食べた] 子供達]に 配らせた]).
  [Mary-NOM John-DAT apple-ACC [[EC (ate] children]-DAT distribute-PASS.PST]).
  "Mary let John distribute apples to the children who ate (it=lunch, etc.)."

- [メアリーが [[[[ジョンに りんごを (あげた] 子供]を 呼び止めた] 女の人]を見た]).
  [Mary-NOM [[[John-DAT apple-ACC give-PST] child]-ACC stop-PAST woman]-ACC see-PST])
  "Mary saw the woman who stopped the child who gave an apple to John."

Yamashita (1997) also notes, crucially, that, due to the lack of constraints imposed by the non-final verbs–for example, the verbs in relative clauses—the first verb does not necessarily determine how the sentence must be completed, as subclauses are also head-final and prenominal in Japanese. The **S NP** structure of relative clauses often are identical to full sentences until the modified NP is observed. Other work suggests that there are argument order preferences in Japanese relative clauses. (Miyamoto and Nakamura, 2013) Furthermore, in previous experiments, Yamashita (1994) finds that sentence completions fitting a simplex structure were preferred for certain ambiguous Japanese sentences, even though simplex structures are not the most common structures in written Japanese, strongly suggesting, Yamashita argues, that criteria aside from word order frequency are used prior to the appearance of the final verb. This leads him to ask what information *is* being used. The three experiments conducted in (Yamashita, 1997) are the following:

- Experiment 1: In a self-paced reading setting, scramble dative-, accusative-, and nominative-marked NPs in Japanese and measure the effect on reading time.
- Experiment 2: Measure the effect of word order vs. case markings on reading time by measuring ditransitive vs. transitive verb reading times given previous context.
- Experiment 3: Measure the specific effects of case marking variety on reading time. A sentence with two nominative-marked words has less variety than a sentence with a nominative and an accusative-marked word.

Experiment 1 revealed no significant difference between scrambled and unscrambled sentences, regardless of whether the scrambled NPs were marked as dative, accusative, or nominative.

Experiment 2 also confirmed that processing time is indifferent to word order, but also that ditransitive verbs are recognized faster than transitive verbs, regardless of the order of the previous arguments.

Experiment 3 found the variety of case markings inversely correlated with reading time.

These results are consistent with the difficulty encountered by frequency and order-based models, such as n-grams, at verb prediction. Moreover, Yamashita suggests that **Experiment 3**'s results might be explained by the increased sentence ambiguity that arises as a result of noun phrases with more than one occurrence of the same case markers. Two nominative noun phrases or two accusative noun phrases, for instance, suggest a subclause, a complex structure which must be then incorporated into the human's parser.

3.1.2 Neurolinguistic Evidence on Verb-final Processing

A complimentary way of analyzing human processing is direct measurement of activity in the brain, and we review studies on linguistic processing of final verbs here.

Electroencephalography (EEG) measures changes in the electromagnetic activity around the brain with very high temporal resolution. Changes in response to stimuli are known as an event-related potentials (ERPS). An ERP is a time-series plot of EEG measurements. An ERP "effect" is an increased response to certain stimuli. To observe ERP effects, researchers average the ERP readings of multiple participants to arrive at a smooth curve. An increase in average deflection immediately post-stimulus is called the "effect". A negative deflection is typically prefixed with an "N"; a positive deflection is prefixed with a "P". For instance, a negative deflection that we will discuss occurring around 400 milliseconds post-stimulus is called the N400. The general consensus among neuroscientists is that the earlier the response to a stimuli, the less taxing the mental processing required to detect the stimuli. Early responses, such as MMN (mismatched negativity, 150-250ms post-stimulus onset) are detected when a pattern deviates in a simple sequence, such as a beep that deviates from other beeps. Other, more complex responses, such as structural (syntactic) and meaning (semantic) deviations occur much later.

For a feature engineering a in a machine learning-based verb prediction system, we must ask what *kinds* of features we expect to be useful. A broad distinction we might draw is that of *semantic* vs. *syntactic* features. For this reason, it is instructive to review studies on human processing of these kinds of features in verb processing.

ERP measurements suggest that both semantic and syntactic information are used when processing verb-final clauses in German, but that they are processed differently (Friederici and Frisch, 2000). The N400 (a negative deflection at about 400ms after stimulus onset) is a well-studied phenomenon usually associated with semantic violations. That is, when a semantic violation is observed, this deflection appears in ERP measurements. This is in contrast to the P600, a positive deflection at about 600ms post-stimulus, which is most often found when humans are presented with a syntactic violation. There are several others, as well, but the N400 and P600 receive significant attention, due to their association with semantics and syntax. These effects occur with most linguistic stimuli, but they are more pronounced when certain violations occur. The measurements for several participants are aggregated pointwise to create smooth time-series curves that can be compared across stimulus conditions. In the following, we will consider studies that observe the N400 and P600: the N400 is typically associated with semantic anomalies; the P600 is associated with syntactic/structural anomalies.

While Yamashita (1997) did not use ERP measurements when studying the effects of case markings and word order on verb processing in Japanese, subsequent studies have used ERP measurements to study the effects of scrambling across languages, and many have failed to find that such scrambling elicits an N400 effect at the verb (Kolk et al., 2003; Hoeks et al., 2004; Chow and Phillips, 2013; Van Herten et al., 2006), even in languages with less free order than Japanese.

Noting this, Chow et al. (2015) use N400 measurements to investigate two competing hypotheses for the initial prediction of an upcoming verb: whether predictions are dependent on all words equally (the bag-of-words hypothesis), or alternatively, whether prediction is selectively modulated by the final verb's arguments (the bag-of-arguments hypothesis). They describe the difference as follows:

> Despite apparent similarities, these hypotheses posit distinct processing mechanisms and have rather different implications for theories or models of linguistic prediction. In particular, it is only under the Bag-of-arguments hypothesis that initial verb predictions are taken to be informed by the clausal structure of the sentence. In order to identify the arguments of the embedded verb, this predictive mechanism must rely on a syntactic representation of the sentence that (at least) marks clause boundaries. In other words, even though the Bag-of-arguments hypothesis posits that verb prediction is initially not sensitive to the structural roles of the arguments, it requires at least a rudimetary chunking of the sentence. On the other hand, under the Bag-of-words hypothesis, the lexical meaning of all preceding words are expected to impact comprehenders' predictions regardless of their structural positions. Therefore, distinguishing between these competing hypotheses will help inform theories about the role of syntactic structure in real-time lexical semantic prediction.

Using the SOV *ba* construction in Mandarin Chinese, they find that the N400 at the verb is modulated by cloze probability when reversing argument roles if the verb is distant from its arguments, and only when the verb is reasonably predictable. (That is, when the entropy is too high, this effect does not appear.)

This distinction is important for several reasons: If minimal chunking is necessary in humans, it may inform inform us of the kind of meaningfuly informative unit we expect in a computational system. Additionally, if humans process words differently based on their arguments (and distance), this gives us reason to believe that (1) certain features may be more useful in a classifier based on their argument structure, and (2) when presenting incremental information to human participants, we should be careful to truncate at meaningful (chunked) boundaries, as opposed to the kind of aggressive tokenization that can occur in morphologically rich languages such as Japanese.

They find, as well, that the arguments that attach to the verb itself initially modulate this effect. With a bag-of-words mechanism, we would expect all preceding words to facilitate this effect, not just those which attach to the verb. More specifically, the bag-of-arguments mechanism works early in the sentence, when comprehenders are initially sorting out which words are arguments and which are not. While we do not do so in this research, for a machine-based verb prediction system, it would likely be useful if we could determine what the likely arguments to the final verb are.

Friederici and Frisch (2000) found that, in German verb-final sentences, the N400 occurred at the verb, as expected, with selectional violations, but unexpectedly also occurred with number-of-arguments (i.e., direct object for verbs that do not take them) violations. They postulate that this could be due to the integration of semantic roles, or that additional object could make for a semantically awkward sentence. Key here is the occurrence of a P600—a positive deflection at 600ms post-stimulus, associated with syntactic anomalies. Based on current understandings of this effect, this result suggests that syntactic and semantic integration is occurring at the verb.

Related experiments have been performed on Japanese. Koso et al. (2011) measured ERPs when modulating the case markings of the arguments in Japanese, also finding both N400 and P600 effects. In Japanese, some verbs strongly prefer datives; others strongly prefer accusatives. They performed two experiments: in Experiment 1, the participants were merely required to read the grammatical and agrammatical sentences; in Experiment 2, the participants were required to make a grammaticality judgment. The researchers found significant differences in the hemisphere of lateralization of the N400, depending on whether an accusative- or dative-preferring verb was encountered, but the amplitudes were not significantly different. It is worth noting that, oddly, the P600 effect was the *opposite* of what is usually observed, being more pronounced when the subjects saw a correct sentence than when they saw an incorrect one. As a possible explanation for this "inverse P600", they speculate that, due to the head-final nature of Japanese, the upcoming sentence completion may be clearer in correct sentences than incorrect ones, since a seemingly-incorrect sentence could be a subordinate clause that has yet to be resolved, leading to a speedup in context-updating. Interestingly, the N400 effect was only observed in the second task, under the condition where a dative verb was mismatched. The LAN effect—occurring around 200ms, often associated with phrase structure building and morphosyntax—occurs with the accusative-preferring verbs. In their words:

> If this general distinction applies to Japanese, our findings in Experiment 2 suggest a very interesting story. The “NP-ga NP-ni + O-verb” [ACC-preferring verb] elicited LAN, while the “NP-ga NP-o + Ni-verb” [DAT-preferring verb] elicited N400. Thus, it may imply that the mismatch (i.e., subcategorization violation) between the “Ni” [DAT] marked NP and “O”-verb [ACC-preferring verb] is indicative of morphosyntactic ill-formedness, while the mismatch between the “O” [ACC] marked NP and “Ni”-verb [DAT-preferring verb] is rather indicative of semantic incongruity. In other words, the requirement of an accusative NP from the “O”-verb [ACC-preferring verb] would be morphosyntactic (or, structural) in nature, while the requirement of a dative NP from the “Ni”-verb [ACC-preferring verb] could be thematic (or, semantic) in nature. The fact that even the same type of syntactic (subcategorization) violation is processed in different manners may indicate the essential difference between the two types of case particles.

(Koso et al., 2011)

The fact that we see divergent neurological responses might suggest that humans are making predictions that are different in kind (or, at least, in relative degree) depending on the case roles observed. As yet, due to the nearly nonexistent literature on machine-based verb prediction, we have no data on whether or how this experiment may translate to a machine-based system. One can imagine, however, a reinforcement learning system that trusts either semantic- or syntax-based predictions more or less *contingent upon* the kind of anticipation.

One computational study (Konieczny and Döring, 2003) did perform a verb prediction study using a Simple Recurrent Network[3] (Elman, 1991) simulation to investigate the relationship between the difficulty of syntactically integrating the verb as a function of the number of arguments. They found that the presence of a dative case marker markedly increased prediction accuracy for the final verb in artificial sentences. These sentences were generated with an constraint grammar consisting of 63 words, including transitive, ditransitive, and intransitive verbs. 15,000 of the sentences were used for training and forty were used for testing. Their results are reported in terms of GPE[4] , a metric peculiar to these kinds of networks, but the clear trend is that the presence of a dative improves the predictions. (Datives reduce ambiguity.) To investigate further,

[3] This is a connectionist formulation of language processing.

[4] Grammatical prediction error. The ratio of the number of non-grammatical output activations to the sum of the total output activations

they performed human experiments with German sentences, measuring reading times with the same sorts of arguments. They found that reading times for embedded verbs with dative-marked arguments were on average 229 ms faster than those marked with a genitive. The types of prepositional phrase (PP) adjuncts had no effect, though a 59 ms average speedup was observed for PPs modifying an NP. This provides evidence for **anti-locality**.[5] As noted in Levy and Keller (2013), while locality effects have been observed in the processing of Russian, English, and Chinese relative clauses, they have been less forthcoming in languages such as Hindi, Japanese (as we saw earlier), and German.

Building on previous work, Levy and Keller (2013) begin by noting how very small differences in a sentence fragment have drastic effects on the possible structural realizations. They use the following example: noting that the change of one character in a case-marked article (from s to m) dramatically changes the sentence structure.

- a. Die Einsicht, dass der Freund **des** Kunden das Auto aus Plastik/Freude

  "The insight that the friend of the-GEN customer sold the automobile made of plastic/out of joy amused the others."

- b. Die Einsicht, dass der Freund **dem** Kunden das Auto aus

  "The insight that the friend sold the customer the automobile (made of plastic/out of joy) amused the others."

We exploit this observation in our feature engineering for German verb prediction.

Levy and Keller (2013) also conducted a corpus study in which they estimate the probability that the next verb is a participle verb, given the prior context in a deep parse. In their first experiment, they found that probability of a participle verb given NP-ACC is 0.546; for NP-DAT + NP-ACC, it is 0.756. This confirms empirically that the DAT+ACC combination constrains the types of verbs that come later. Furthermore, they found that a dative in the main clause led to greater expectation among human participants. In their second experiment, they embedded the clauses used in the previous experiment, making them relative clauses. They

[5] Briefly, a **locality** effect occurs when the presence of competing modifying constituents in sentence input increases processing load. An **anti-locality** effect occurs when their presence facilitates, rather than hinders, processing.

found the conditional probabilities to be much lower. NP-ACC predicts a participle verb with probability .073, while NP-DAT + NP-ACC predicts with probability 0.103. Despite the decreased overall probabilities, there is evidence for facilitation by an adjunct PP that was absent in the previous experiment. Unlike in the previous experiment, in the case of a relative clause, NP-PP + NP-ACC appears to facilitate prediction (probability .103, significant for $p < 0.01$). The counts of preverbal datives in this second experiment were low, however (48 examples).

> [M]oving the adjunct phrase into the [relative clause] does not drive up reading times when the dative NP is in a preceding subordinate clause, but it does when the dative NP is a preverbal dependent inside the [relative clause].

The key point of these experiments is that both locality and anti-locality effects can occur in the same structure. The increased difficulty when both and adjunct PP and a dative NP were present in relative clauses indicates a locality effect. From the standpoint of machine-based prediction, the big picture is that we must be careful. Arbitrarily adding features to a classifier, for instance, may not be the best approach. There is substantial evidence that structures such as relative clauses can confound predictions of the final verb. Abekawa and Okumura (2005) were able to identify relative clauses (consisting of an **S NP** structure) in Japanese corpora with support vector machines, even distinguishing between head-restrictive relative clauses and case slot-gapping relative clauses, despite the fact that these are syntactically identical in Japanese. An incremental version of this could be a useful feature the system a verb prediction and/or simultaneous translation system, but this requires access to labeled relative clause data, which is unavailable at the present time.

Other more tangental recent work has gone into predicting verb clusters (Peterson et al., 2016) in English with VerbNet (Schuler, 2005) and mixture models, as well as incuding semantic frames for verb classes from large corpora (Kawahara et al., 2014) Other work has examined transforming sentences to agree with an equivalent verb (Kaji et al., 2002), which could be useful for correction in a translation system: if a mistake is made, try to find a paraphrase that preserves the meaning and agrees with the case roles that have already been utterred.

3.2 Human Verb Prediction

Having covered relevant background information on verb prediction, we proceed to describe our studies on incremental verb prediction. We first examine human verb selection in a constrained setting to better understand the kind of performance we should demand of computational approaches. While we know that humans make incremental predictions across sentences, we do not know how skilled they are in doing so, and while it is possible that machines, with unbounded memory and access to Internet-sized data, could theoretically outperform humans, this study allows us to appropriately gauge our expectations for computational systems.

We use crowdsourcing to measure how well novice humans can predict the final verb phrase of incomplete Japanese sentences in a multiple choice setting. We use Japanese text of the Kyoto Free Translation Task corpus (Neubig, 2011, KFT), a collection of Wikipedia articles in English and Japanese, representing standard, grammatical text and readily usable for future SOV-SVO machine translation experiments.

3.2.1 Extracting Verbs and Sentences

This section describes the data sources, preparation, and methodology for the crowdsourced verb prediction task. Given an incomplete sentence, participants select a sentence-final verb phrase containing a verb from a list of four choices to complete the sentence, one of which is the original completion from the corpus.

We randomly select 200 sentences from the development set of the KFT corpus (Neubig, 2011). We choose these data because the sentences are from Wikipedia articles and thus represent widely-read, grammatical sentences. These data are directly comparable to our computational experiments and readily usable for future SOV-SVO machine translation experiments.

We ask participants to predict a "verb chunk" that would be natural for humans. More technically, this is a sentence-final **bunsetsu**. A *bunsetsu* is a commonly used linguistic unit in Japanese, roughly equivalent to an English phrase: a collection of content words and zero or more functional words. Japanese verb *bunsetsu* often encompass complex conjugation. For example, a verb phrase 読みたくなかった(read-DESI-NEG-

PAST), meaning ‘didn’t want to read’, has multiple tokens capturing tense, negation, etc. necessary for translation. We identify verb *bunsetsu* with a dependency parser (Kurohashi and Nagao, 1994). Of interest are *bunsetsu* at the end of a sentence that contain a verb. We also use *bunsetsu* for segmenting the incomplete sentences we show to humans, only segmenting between *bunsetsu* to ensure each segment is a meaningful unit.

Answer Choice Selection

We display the correct verb *bunsetsu* and three incorrect *bunsetsu* completions as choices that occur in the data with frequency close to the correct answer in the overall corpus. We manually inspect the incorrect answers to ensure that these choices are semantically distant, i.e., excluding synonyms or troponyms.

Sentence Presentation

We create two test sets of truncated sentences from the KFT corpus: The first, the **full context set**, includes all but the final *bunsetsu*—i.e., the verb phrase—to guess. The second set, the **random length set**, contains the same sentences truncated at predetermined, random *bunsetsu* boundaries. The average sentence length is nine *bunsetsu*, with a maximum of fourteen and minimum of three. We display sentences in the original Japanese script.

Participants view the task as a game of guessing the final verb. Each fragment has four concurrently displayed completion options, as in the prompt (2) and answers (3). Users receive no feedback from the interface.

We use CrowdFlower[6] to collect participants’ answers, at a total cost of approximately USD$300. From an initial pool of fifty-six participants, we remove twenty via a Japanese fluency screening. To ensure the validity of this screening, we test it on non-native but highly proficient Japanese learners. None passed this screening. In the crowdsourcing task, from each participant we collect guesses for five sentences.

(2) 谷崎潤一郎は 数寄屋を

Junichiro Tanizaki-TOP tea-ceremony house-OBJ

(3) (a) 好んだ

like-PAST

[6] http://www.crowdflower.com/

(b) 変えられた

change-PASS CAP-PAST

(c) 始まったとされている

begin-PAST-COMP-suppose-AUX.PRES

(d) 増やしていた

increase-AUX.PAST

3.2.2 Presenting Partial and Complete Sentences

The first task, on the **full context set**, informs us of the ways in which humans predict the sentence-final verb chunk with *all* context—everything but the final verb *bunsetsu*— available. The second task, on the **random length set**, shows how the amount of revealed data affects the predictability of the final verb chunk. We examine a correlation between the length of the pre-verb sentence fragment and participants' accuracy (Figure 3.1).

Psycholinguistic experiments using lexical decision tasks suggest Japanese speakers being syntactic processing by using case—the type and number of case-marked arguments—before the verb's availability (Yamashita, 2000). We also examine here the correlation between the number of case markers[7] and accuracy. It is likely that the number of case markers and the length of the sentence fragment are confounded; so, we create a measure, the proportion of case markers to the overall sentence information (the number of case markers in the fragment divided by the number of *bunsetsu* chunks). We call this **case density**.

3.2.3 Results of Human Experiments

We now discuss the results of our crowdsourced human experiments. In the **full context set**, average accuracy over 200 sentences is 81.1%, significantly better than chance ($p < 2.2 \cdot 10^{-16}$). Figure 3.1 shows the accuracy per sentence length as defined by the *bunsetsu* unit. A one-way ANOVA reveals a significant effect of the sentence length ($F(1, 998) = 7.512, p < 0.00624$), but not the case density ($F(1, 998) = 1.2, p = 0.274$).

In the **random length set**, average accuracy over 200 sentences is 54.2%, significantly better than

[7] In this study, we counted case markers that mark nominative (-*ga*), accusative (-*wo*), ablative (-*kara*), and dative (-*ni*).

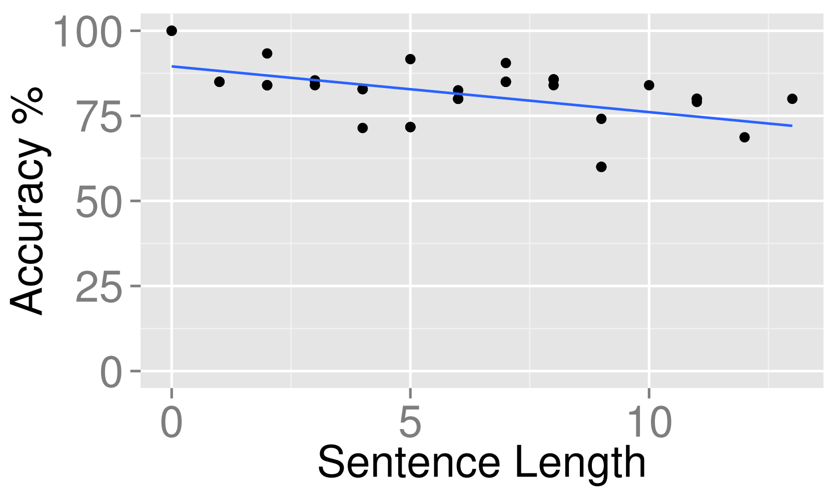


Figure 3.1: Full context set: Accuracy is generally high, but slightly decreases on longer, more complicated sentences, averaging 81.1%.

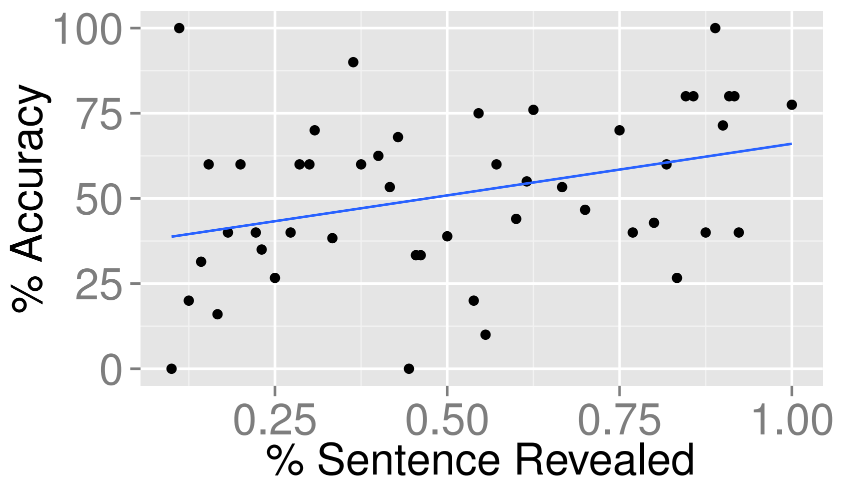


Figure 3.2: Random length set: The accuracy of human verb predictions reliably increases as more of the sentence is revealed.

chance ($t(199) = 11.8205, p < 2.2 \cdot 10^{-16}$). Figure 3.2 shows the accuracy per percentage of length of the presented sentence fragment. A one-way ANOVA reveals a significant effect of the sentence length ($F(1, 998) = 57.44, p < 7.94 \cdot 10^{-14}$). We also find a significant effect of the case density ($F(1, 998) = 5.884, p = 0.0155$).

3.2.4 Discussion

The reuslts indicate that predictability increases with the percent of the sentence available in all of our experiments. While this is hardly a surprising result, it is important for gauging our expectations for a computational system: if a system's accuracy vascillates wildly, for instance, we should investigate why.

By the end of the sentences, the verb chunks are highly predictable by humans in the multiple choice setting. Participants choose the final verb more accurately as they gain access to more case markers in the **random length set** but not in the **full context set**.

Case density is a significant factor in predictive accuracy on the **random length set** for humans, suggesting that case is more helpful in predicting a sentence-final verb when the preceding contextual information is insufficient. The following example illustrates how case helps in prediction. The nominative and accusative markers greatly narrow the choices, as shown in (4).

A recent psycholinguistics study on incremental Japanese verb-final processing (Momma et al., 2015) argues that native Japanese speakers plan verbs in advance, before the articulation of object nouns, but not subject nouns. Since case markers assign the roles of subject and object in Japanese, we expect that a high ratio of case markers to words will increase predictability of verbs. In addition, Yamashita (1997) argues that the *variety* of case markers increases predictability just before the verb. Our results further support the proposition that case markers modulate predictability in SOV verb-final processing.

(4) 江戸幕府区-**が** 成立すると 寺院法度-に-より —

Edo shogunate-**NOM** establish-do-**CONJ** temple-prohibition-etc.-ACC-for
'After Edo shogunate has established, due to the temple prohibition etc. —'

In other cases, there exist choices, which, while incorrect, could naturally complete the sentence. These questions are frequently missed. For instance, in one 90% revealed sentence, the participant has the choices: (i) 収める (put-PRES), (ii) 厳しくなる (strict-become), (iii) 収録されている (record-do.PASS-AUX.PRES), and (iv) 務める (work-PRES). Choice (i) is the correct answer, but choice (iii) is a reasonable choice for a Japanese speaker. All participants missed this question, and all chose the same wrong answer (iii). We leave a cloze task where participants can freely fill in the sentence-final to future work.

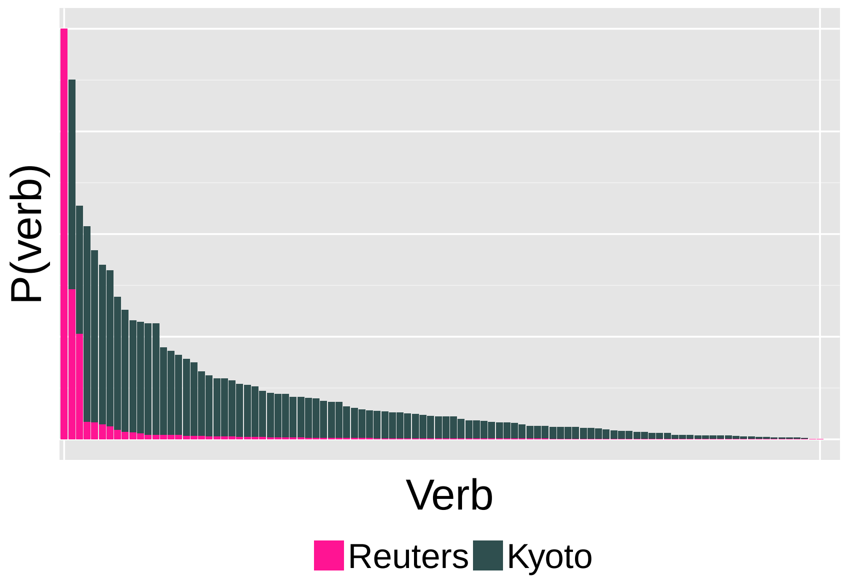


Figure 3.3: Distribution of the top 100 content verbs in the Kyoto corpus and the Reuters Japanese news corpus. Both are Zipfian, but the Reuters corpus is even more skewed, even with the common special cases excluded.

These results provide a basis of comparison for automatic prediction. In the next section, we examine whether computational models can predict final verbs and compare the models' performance to that of humans.

3.3 Machine Verb Prediction

Now that we have the results of human verb predictability, we have baselines against which we may compare computational verb prediction approaches. In this section, we introduce incremental verb classification with a linear classifier.[8]

For our investigation of computational verb classification, we use two very different languages that both have verb-final syntax—Japanese, which is agglutinative, and German, which is not—and show that discriminative classifiers can predict final verbs with increasing accuracy as more context of sentences is revealed.

[8] While we use logistic regression, using hinge loss achieves similar accuracy.

The first approach to incremental verb prediction applied to German (Chapter 4) achieves poor accuracy. This approach creates a Kneser-Ney n-gram language model for the prior context associated with each verb in the corpus; i.e., 50 n-gram models for 50 verbs. Given pre-verb n-gram context c in a sentence S_t, and verb prediction $v^{(t)} \in V$, the verb selection is defined by the following equation:

$$v^{(t)} \equiv \arg\max_{v} \prod_{c \in S_t} p(c \,|\, v) p(v). \tag{3.1}$$

Then, it chooses the verb that maximizes the probability of the observed context, scaled by the prior probability of the verb in the overall corpus. Unsurprisingly, given the distribution of verbs in real data (Figure 3.3), this n-gram-based approach has low accuracy and tends to predict the most common verb in each language. For a system attempting to determine whether or not these predictions should be trusted, this degenerates into the less interesting problem of whether to trust whether the final verb is indeed the most common one. While this improves translation delay on average, better predictions will lead to more significant improvements. We instead opt for a one-vs-all discriminative classification approach.[9]

3.3.1 Discriminative Classificaiton

For two random variables, X and Y, a generative model, such as a typical n-gram language model must approximate the full joint distribution $P(X, Y)$. In the language model-based verb prediction approach, this requires that we approximate the prior distribution of the verbs, $P(V)$. Due to the distribution of verbs, however, giving equal weight to this prior distribution heavily skews our predictions toward the most common verbs. In discriminative classification, we only approximate the conditional distribution, $P(Y|X)$. For verb prediction, we approximate the probability of the verb given the context *directly*, $p(v|c)$, where c is a particular word context and v is the verb whose probability we are estimating. In cases where the prior probability is heavily skewed but also highly context dependent, this can yield more varied and informative estimates.

[9] One-vs-all classification builds a classifier for each class versus the aggregate all other classes.

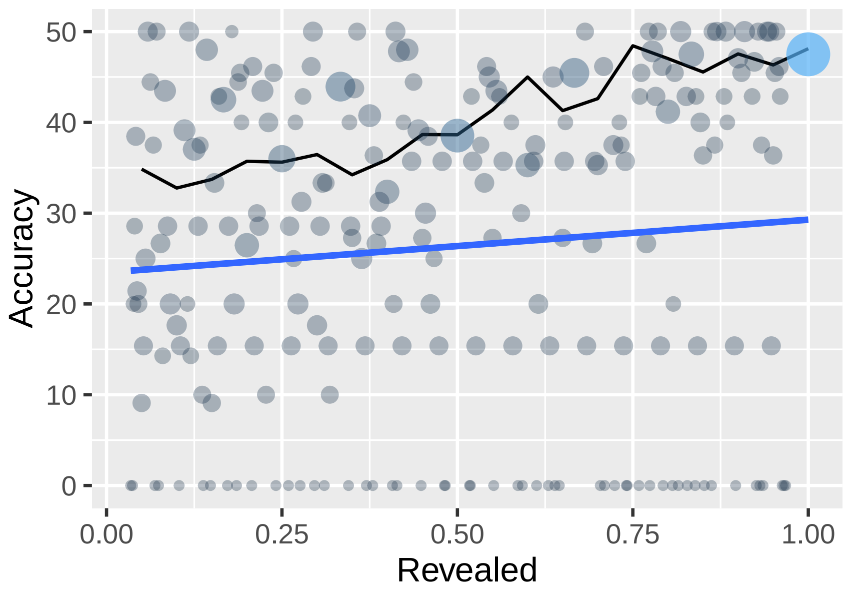


Figure 3.4: Verb classification results on crowdsourced sentences. Despite many out-of-vocabulary items and significant noise, the average accuracy, shown in the non-monotonic line in the plot, increases over the course of the sentence. Larger, darker circles indicate more examples for a given position. Accuracy was calculated by aggregating the guesses at 5% intervals.

3.3.2 One-vs-all Classification

One-vs-all classification (also known as one-vs-rest) is a common heuristic reduction for multiclass classification problems. This scheme creates a binary classifier for reach label—in this case, each verb—called a base classifier. For a base classifier c_i with the label i, the positive class $(+1)$ is the label i and the negative class (-1) is the label $\bar{i}$, representing any label other than i. At training time, the approach samples for class $\bar{i}$ from the set of examples that do not have the label i. Thus, for 50 labels, we have 50 base binary classifiers. At prediction time, all 50 base classifiers are queried with the same feature vectory, yielding a confidence score. In this case of logistic regression, this is a probability. The label which has the base classifier with the highest score for $+1$ is chosen as the prediction.

3.3.3 Classification on Human Data

We first incrementally classify verbs on the same 200 sentences from the human experiments in Section 3.2. Since the answer choices are often complex verb *bunsetsu* and since many of these verb phrase answer choices do not appear among the most common verbs, lemmatizing the verbs and performing one-vs-all classification yields extremely low accuracy. Thus, we use binary classification with a single linear classifier to produce a probability for each candidate answer, encoding the verb phrase itself into the feature vector. While there are some similarities to the error-correcting output code (ECOC) multiclass classification method—to the extent that it is a reduction to binary classification—this method is simpler.

3.3.4 Training a Morphological Model

The processing is as follows: We train on 463,716 verb-final sentences extracted from the training data. We use both **context features** and **final verb features**. Our context features, i.e., those preceding the final verb, are represented as follows: the context **unigrams** and **bigrams** take a value of 1 if they are present and 0 otherwise; **case markers** observed in the sentence context are represented as unigrams and bigrams in the order that they appear; and we reserve a distinct feature for the **last observed case marker** in the sentence. Our **verb features** consist of the final verb's tokens given by the morphological analyzer, which, in addition to the verb stem itself, typically include tense and aspect information. These are represented as unigrams and bigrams in the feature vector.

To allow the classifier to learn, we must encode the interactions between the verb features and the context features. Thus, we use the Cartesian product of sentence and verb features to encode interactions between them: for each training sentence we generate both a positive and a negative example. The example with the correct verb phrase is labeled as a positive example ($+1$), and we uniformly select a random verb phrase from one of the 500 most common verb phrases and label it as negative (-1) example for the same sentence context, yielding 927,432 training examples and 267,037,571 features. While we experimented with several numbers of weighted negative examples, we found that one negative example with of equal weight to the positive gave the best results of the configurations we tried.

For clarity, we describe this feature representation more formally. Given sentence S_t with a pre-verb context consisting of unigrams, bigrams, case marker tokens,$C = \{c_0, ..., c_n\}$, and *bunsetsu* verb phrase tokens $A = \{a_0, ..., a_k\}$, the feature vector consists of $C \times A = \{c_0 \wedge a_0, c_0 \wedge a_1, ..., c_n \wedge a_k\}$, where $\wedge$ concatenates the two context and answer strings. During learning, the weights learned for the concatenated tokens are thus based on the relationship between a context token and a *bunsetsu* token and mapped to $\{+1, -1\}$. More concretely, individual morphemes of the Japanese verb phrase are combined with the pre-verb unigrams, bigrams, and uniquely identified case marker tokens. We find that accuracy improves when the morphemes used in the negative examples and positive examples are disjoint; so, we enforce this constraint when selecting negative examples. For example, if the positive example includes the past tense morpheme, た, the negative example is altogether disallowed from having this morpheme as a verb feature.

This is useful because we can use the semantically meaningful decomposition of the verbs to form classifiers with features that represent them without opting requiring an explicitly hierarchical model. The classifier implicitly learns to consider, for example, how likely a sentence is to have a past tense verb, a negated verb, a passive verb, and so on. This would not be feasible in a one-vs-all classification scheme. This allows, furthermore, a measure of coherence of not only the case markers and the verbs, but the case markers, the verb morphemes (and all the they represent), the words themselves, and the verbs.

3.3.5 Choosing an Answer

At test time, we test progressively longer fragments of each sentence, extracting the aforementioned features online until the entire pre-verb context is available. We proceed from beginning to end, one token at a time. Each query to the classifier is independent. For every sentence fragment, the classifier determines the probability of each of the four possible verbs by adding their verb features to the feature vector of the example. The answer choice with the highest probability of $+1$ is chosen as the answer. By taking this approach, we can model complex verbs and their context jointly. Intuitively, the probability of a $+1$ is the model's prediction of how well the *bunsetsu* verb phrase coheres with the sentence context. That is, a higher probability of $+1$ indicates that all of these different kinds of features (unigrams, bigrams, case markers, answers) tend to occur together.

A potentially confusing aspect of this encoding scheme concerns the fact that the answers themselves are in the feature vector. But this is true for *all* answers. We create queries for every possible label. The label that best coherest with the context is selected.

Some verbs are absent from the training data, forcing the classifier to rely on morphemes to distinguish between them. The alternative—e.g., in a typical one-vs-all classification approach—is that the classifier could reason from nothing whatsoever when a fully-inflected verb is absent from the training data. Given the complexity of *bunsetsu*, this happens often even in large corpora for a language such as Japanese.

3.3.6 Multiple Choice Results

Despite having only four choices from which to choose, this task is in many ways more difficult than the 50-label classificaion problem described in the next section. This is due to the added complexity inherent to modeling the effect of morphemes and missing examples. These limitations notwithstanding, the accuracy does improve as more of the sentence is revealed (Figure 3.4), indicating that the algorithm learns to use these features to rank verbs, though the performance significantly lags that of both the human participants and our later experiments.

Additionally, on the **full context set**, sentence length is negatively correlated with accuracy (Figure 3.5), as in the much more convincing results of our human experiments (Figure 3.1), though the trend is not entirely consistent, making it difficult to draw firm conclusions. Case density is again positively correlated with accuracy on both the random (Figure 3.8) and full context sets.

An Illustrative Example

To gain some insight into how features can influence the classifier, we here examine an example of the classifier's behavior on the multiple choice data.

(5)

少年時代-は	熊本藩-の	藩校-で	儒学-を
childhood days-TOP	Kumamoto domain-GEN	clan school-LOC	Confucianism-ACC
学び、	後-に	西本願寺-において	修行-に
study-MED	subsequently-LOC	Nishihongan Temple-LOC	discipline-ALL

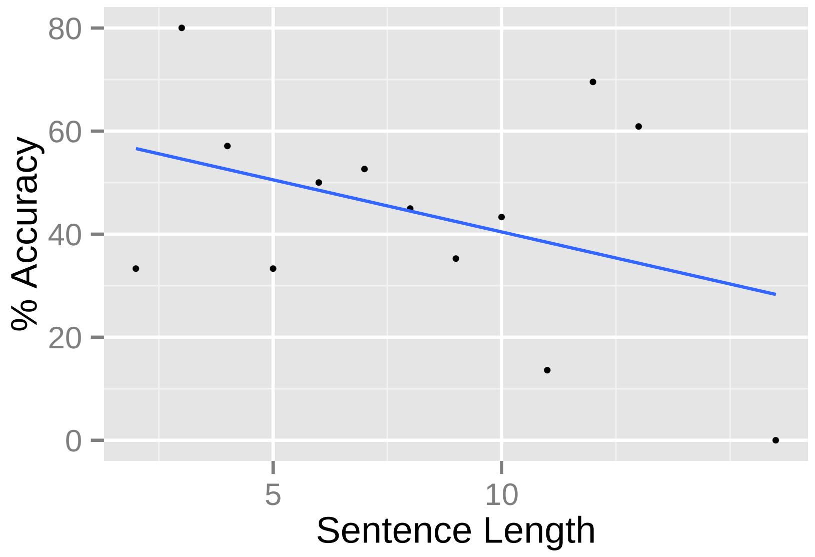


Figure 3.5: Classification accuracy as a function of sentence length on the full context set. While there is a clear correlation between sentence length and accuracy, there are several outliers. Compare to Figure 3.1.

(6) (a) 励ん-だ

strive-PAST

(b) 創刊-さ-れ-る

issue-do-PASS-NPST

(c) 加え-られ-てい-る

add-PASS-CONT-NPST

(d) 勤め-る

serve-NPST

In Example (5), the classifier incorrectly chooses “issue” as the verb until observing the accusative case marker attached to “Confucianism”. At this point, the classifier’s confidence in the correct answer rises to 0.74—and correctly chooses “strive”. This answer goes unchanged for the remainder of the sentence, though “study” attaches to “Confucianism”, not the final verb. The combined evidence, however, is enough for the classifier to select correctly, and indeed, most of the following tokens only increase the classifier’s confidence. Adding “subsequently” increases confidence to 0.84, an intuitive increase given the likely tense information contained in such a word. The somewhat redundant case marker here only increases confidence to 0.86.

Adding the reference to the temple decreases confidence again to 0.79. But adding the **final case marker**, which also forms a new bigram with the previous word, results in a huge increase in confidence, to 0.90. This is shown visually in Figure 3.6. Additionally, we can see in Figure 3.7, which shows the total change in confidence from the previous time step, that case markers are particularly informative in this example.

3.3.7 Multiclass Verb Prediction

While the multiple choice experiment was more open-ended (predicting random verbs), we now focus on a more constrained task: how well can we predict the most frequent verbs. If we can improve upon this, we can improve simultaneous translation. We showed a slight improvement in simultaneous translation by using n-gram language model-based verb prediction in Chapter 4. We show a large improvement over their approach to verb prediction using a discriminative multiclass logistic classifier (Langford et al., 2007).

Data Preparation

Our classes for multiclass classification are the fifty most common verbs in the KFT (Japanese, as in the human study) and Wortschatz corpora (Biemann et al., 2007, German).

We use data from the training and test sets of the KFT Japanese corpus of Wikipedia articles and a random split of the German Wortschatz web corpus, from which we extract the verb-final sentences. In Chapter 4, we used an n-gram model to distinguish between the fifty most common German verbs for SOV-SVO simultaneous machine translation, which we replicate as our baseline. Following this study, we train a model on the fifty most common verbs in the training set.

In Japanese, due to the small size of the standard test set, we split the data randomly, training on 60,926 verb-final sentences ending in the top fifty verbs and testing on 1,932. Our total feature count is 4,649,055. We use the MeCab (Kudo, 2005) morphological analyzer for segmentation and verb identification. We consider only verb-final sentences. We skip semantically vacuous post-verbal copulas when identifying final verbs.

Finding Verbs

We identify verbs in the German text with a part-of-speech tagger (Toutanova et al., 2003) and select from the top fifty verbs. We consider the sentence-ending set of verbs to be the final verbs. We train on

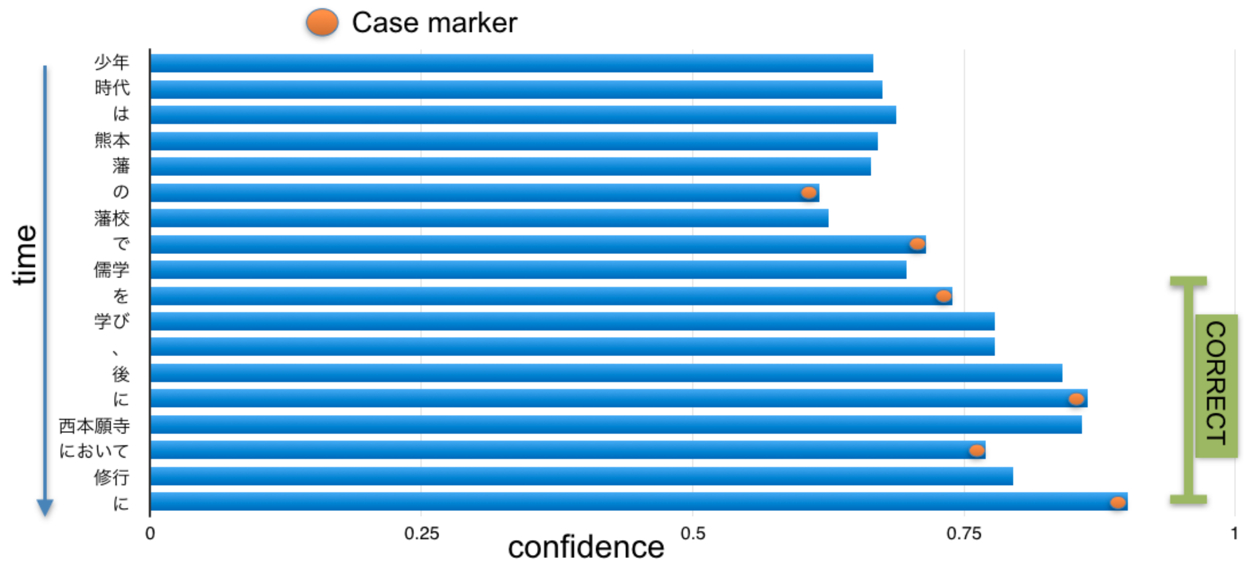


Figure 3.6: Classification confidence across the sentence. The confidence increases as more context is revealed. Case marker lead to substantial changes in confidene.

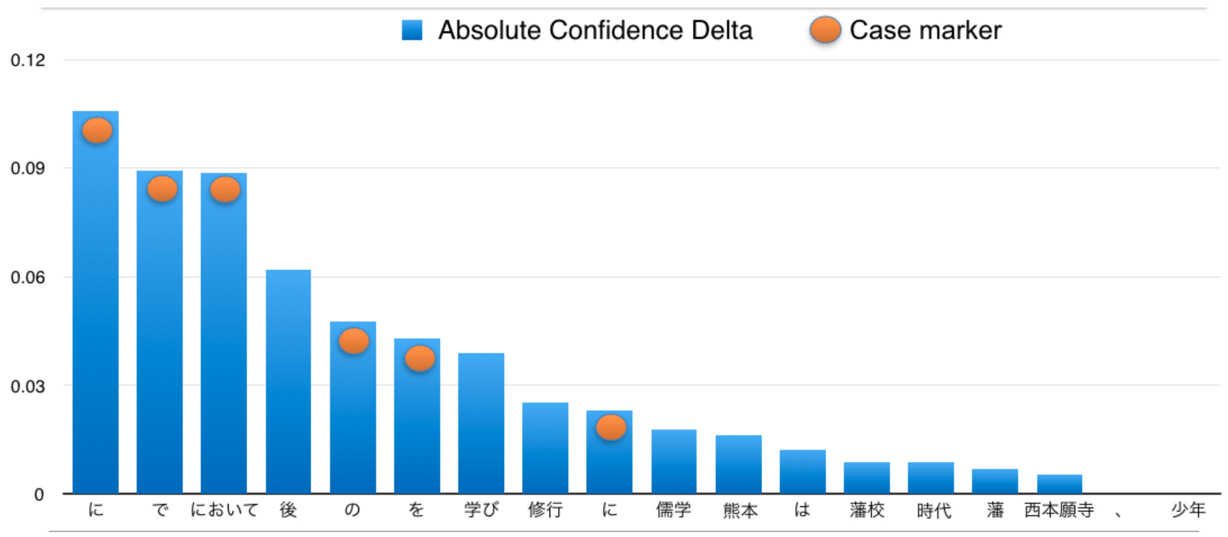


Figure 3.7: Sentence tokens sorted by the absolute value of the change in confidence from previous time step. Case markers, marked with orange circles, are responsible for most of the largest swings in confidence in this example.

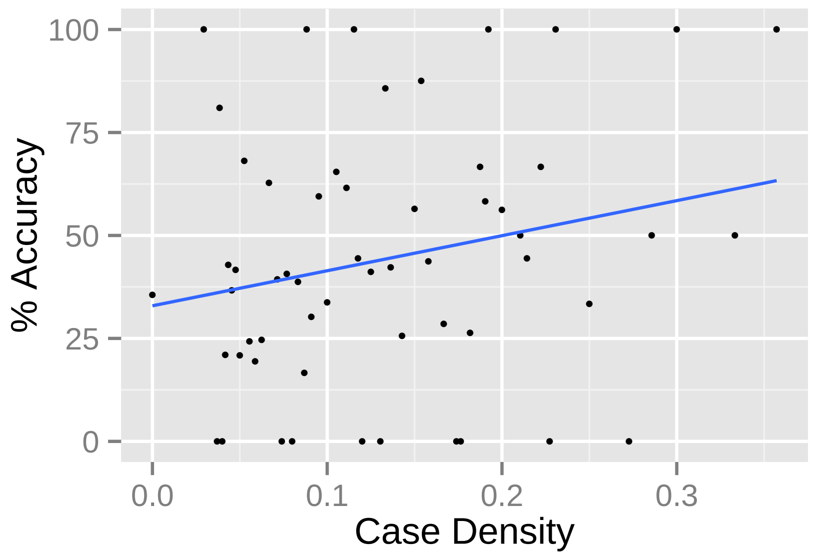


Figure 3.8: Classification accuracy as a function of case density on the incremental sentences. The accuracy is correlated with case density, but the data are extremely noisy. Full-context accuracy has a similar trend (not shown).

76,209 verb-final sentences ending in the top fifty verbs and test on 9,386. In German, to approximate the case information that we extract in Japanese, we test the inclusion of equivalent unigram and bigram features for German **articles**, the surface forms of which determine the case of the next noun phrase.

In Japanese, we omit some special cases of light verbs that combine with other verbs, as well as ambiguous surface forms and copulas.[10]

Features

All features are encoded as binary features indicating their presence or absence. For Japanese, we again include **case unigrams**, and **case bigrams**, which encode as distinct features the for case markers observed thus far.[11] We also include a feature for the **last observed case marker**. For both Japanese and

[10] In Japanese, we omit some ambiguous cases and variants of “is” and “do”: excluded are variants of *suru* (“to do”), which combines with nouns to form new verbs, *aru* (“is”, inanimate case), and *iru* (“is”, animate case). The tokens *aru* and *iru* also combine with other verbs to change tense and aspect, in which case they are not verbs, and can form the copula *de aru*. Distinguishing between all of these cases is beyond the scope of this study; so, they are excluded. We also omit duplicates that are spelled differently (i.e., the same word but spelled without Chinese (*kanji*) characters and slightly different forms of the same root).

We also omit the light verb *naru* (“to become” or “to make up”) for similar reasons to *suru*. The increasing trend shown in the results does not change with their inclusion.

[11] For instance, given a sentence fragment X-に Y-を, representing X-DAT Y-ACC, the case bigram would be に∧を.

German, we normalize the verbs to the non-past, plain form, both providing more training data for each verb and simplifying the job of our classifier.

German case is conveyed primarily through articles and pronouns, so we include special features for articles. For example, for the sentence “Es wurde ihnen von einem alten Freund geholfen”, we add the features `ART_es_ihnen` and `ART_ihnen_einem` to convey case information beyond individual words and bigrams.

Individual tokens are also used as binary features, as well as token bigrams.

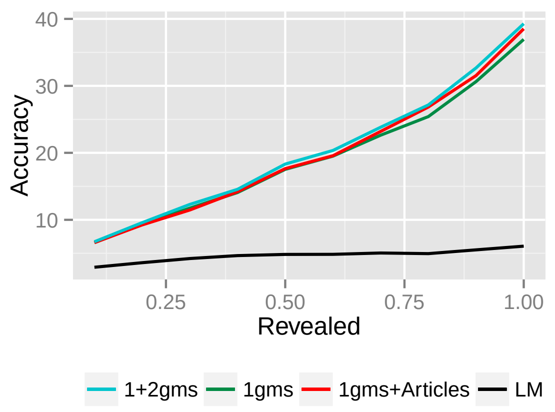
An Example for Every Word

In a simultaneous interpretation, a person or algorithm receives a constant stream of words, and each new word provides new information that can aid in prediction. Previous predictive approaches to simultaneous machine interpretation have taken this approach, and we also use it here: as each new word is observed, we make a prediction. This is a generalization of **random** presentation of prefixes in the human study.

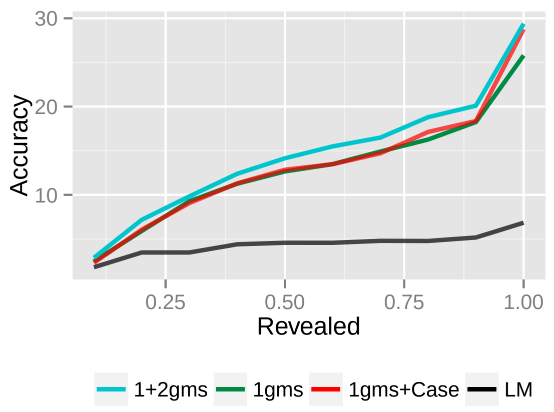
3.3.8 Classification Results and Discussion

Better at the End

The discriminative approach greatly outperforms the n-gram classifier, which has a tendency to over-predict frequent verbs. By the end of the sentence, accuracy reaches 39.9% for German (Figure 3.9a) and 29.9% Japanese (Figure 3.9b), greatly exceeding choosing the most frequent class baseline of 3.7% (German) and 6.05% (Japanese). The n-gram language model also outperforms this baseline, but not by much. It also improves over the course of the sentence, but the model cannot reliably predict more than a handful of verbs in either language.



(a) German average prediction accuracy over the course of sentences. Bigrams help slightly in the second half of the sentence. Adding special features for case-assigning articles to unigrams nearly matches the performance of adding all bigrams in the final 10%. All handily outperform the trigram language model.



(b) Japanese average prediction accuracy over the course of sentences. Adding bigrams consistently outperforms unigrams alone in Japanese, possibly due to the agglutinative nature of the language. The accuracies diverge the most toward the end of the sentences: Adding only explicit case markers to unigrams nearly matches performance of adding all bigrams toward the end. All outperform the trigram language model.

Figure 3.9: Classification results on Japanese and German sentences.

Richer Features Help (Mostly at the End)

Bigram features help both languages, but in Japanese more so than in German. Beyond bigrams, however, trigrams and longer features overfit the training data and have a deleterious impact on performance. The superior performance for Japanese bigrams is likely because word boundaries are not well-defined in Japanese, and individual morphemes can combine in ways that significantly add information. German word boundaries are more precise and words (particularly nouns) can carry substantial information themselves. While this might be surprising, given the almost completely free word order of Japanese (aside f rom the head-finality), the aggressive tokenization likely makes the n-grams more effective.

Richer features make more substantial contributions toward the end of the sentences. In Japanese, the addition of bigrams consistently outperforms unigrams alone, but in both Japanese and German, adding special features for tokens with case information helps almost as much as adding the full set of bigrams. In Japanese, case markers always immediately follow the words marked, and in German the articles precede the nouns to which they assign case; thus, rather than relying on isolated unigrams, using bigrams provides opportunities to encode case-marked words that more narrowly select for verbs. In Japanese, the differences are more pronounced toward the very end of the sentences (and less so in German).

Richer features help more at the end, but not merely because the last words of the sentence represent the densest feature vectors. In Japanese, the last word is usually a case-marked noun phrase or adverb that attaches to the main predicate. The final word is therefore immune to subclause interference and must modify the final verb, boosting the classifier performance in these final positions and amplifying the predictive discrepancies between the various feature sets. Accuracy spikes at the end of Japanese sentences, where case information helps nearly as much as adding the entire set of bigrams, further supporting the notion that case information is crucial.

In it conceivable that deeper processing—e.g., separating case-marked words in subclauses from those in the main clause—would likely be more useful, but achieving this incrementally is non-trival.

There were several features an feature-selection strategies that we tried which had negligible impact of classifier performance. It is, however, possible that they can be used in some way to aid performance; thus, they are described here to aid future research efforts.

- **Adding only case marker unigrams (instead of bigrams)** This has no impact on accuracy.
- **Fltering the features by using only case-marked words** This significantly hurt performance, as too many sentence information was omitted. This excludes, for example, adjectives, non-sentence final verbs, and a number of adverbs.
- **Only allowing one word per case marker in the feature vector (the most recent)** The intuition behind this strategy is that the proximity to the verb is indicative of its semantic importance and likelihood of syntactic attachment to the final verb. For example, if a sentence has two accusative particles, the first is likely inside of subclause, whereas the most recent (the one closest to the verb) is more likely to attach to the final verb. Thus, there is some reason to believe that "forgetting" the

earlier words with the same case marker might reduce the confusion of the classifier. This was not the case in our experiments.

- **Using decaying weights on features further in the past** This is an extension of the approach listed above, with more nuance. It did not yield any noticable improvements.

- **Adding part-of-speech tag n-grams** In much the same manner as we did with case markers, this added n-grams of the part-of-speech tags, based on their order in the sentence.

- **Adding the word nearest to the centroid of the observed context in a word embedding space.** We trained a simple word2vec Mikolov et al. (2013) model on the same training data as the classifier and calculated the nearest n words to the centroid of the observed words to the feature vector of the classifier at test time. The intuition behind these features is that the word embeddings might be able to "intuit" information not yet revealed. While these features may have potential, they did not lead to significant increases in accuracy in our experiments.

3.4 Conclusion

Verb prediction is hard for both machines and humans but impossible for neither. Verbs become more predictable in discriminative settings as more of the sentence is revealed, and when all of the prior context is available, the verbs are highly predictable by humans when a limited number of choices is available, though even then not perfectly so. While we make no claims concerning upper or lower bounds of predictability in different settings, our dataset provides benchmarks for future verb prediction research on publicly available corpora: cognitive scientists can validate prediction, confusion, and anticipation; engineers have a human benchmark for their systems; and linguists can conduct future experiments on predictability.

We find that shallow features can be used to predict verbs more accurately with more context. Improving verb prediction can benefit simultaneous translations systems that have already shown to benefit from verb predictions, as well as enable new applications that involve predicting future linguistic input.

In the future, we would like to further explore the potential for richer features, especially predicate argument structures. Intuitively, we want to match the correct predicates (in this case, verbs) to their arguments, but we train on entire sentences. Given the availability of predicate-argument analyzers (Watanabe et al., 2010), dependency parsers in Japanese (Uchimoto et al., 1999; Flannery et al., 2012) and Germans (Nivre et al., 2006; Sennrich et al., 2009), and predicate-argument annotated corpora Burchardt et al. (2006) in German, it is feasible to train models to predict missing predicates incrementally.

In the following chapter, we address the learning problem in a simultaneous translation system and demonstrate both that predictions in a simultaneous translation system can improve performance when they are used correctly, and that an idealized translation system can learn when to use such predictions to perform simultaneous translations. We then proceed from an idealized translation system with rudimentary predictions to a more realistic system (Chapter 5) that can realistically run in real-time, incorporating our verb prediction results from this chapter.

Chapter 4

Reinforcement Learning for Simultaneous Machine Translation

4.1 Introduction

Having explored the prediction of final verbs, we now introduce a simultaneous machine translation system that predicts unseen verbs and uses reinforcement learning to learn when to trust these predictions and when to wait for more input.[1] First, we describe a conceptual system that uses an *omniscient translator* and a simple verb prediction scheme. This allows us to abstract away the learning problem from translation issues and decoding. It also demonstrates that our system performs well even when verb predictions are very poor.

Later, we the incorporation of more realistic components (Chapter 5)—including our improved verb prediction system and (Chapter 3) a state-of-the-art translation system—into our framework and demonstrate that our approach generalizes even without idealized, abstract components.

Simultaneous translation is the production partial translation of a sentence before the input sentence is complete, and is often used in important diplomatic settings. One of the first noted uses of human simultaneous interpretation was the Nuremberg trials after the Second World War. (Gaiba, 1998) Siegfried Ramler (Ramler and Berry, 2009), the Austrian-American who organized the translation teams, describes the linguistic predictions and circumlocutions that translators would use to achieve a tradeoff between translation latency and accuracy. The audio recording technology used by those interpreters sowed the seeds of technology-assisted interpretation at the United Nations (Gaiba, 1998).

Performing real-time translation is especially difficult when information that comes early in the

[1] This is based on work that first appeared in Grissom II et al. (2014), for which I was responsible for all verb prediction-related work. I also contributed to data preprocessing, debugging, experimental design, figure generation, and of course the majority of the writing.

ich	bin	mit	dem	Zug	nach	Ulm	**gefahren**
I	am	with	the	train	to	Ulm	**traveled**
I	(...... **waiting**......)					**traveled** by train to Ulm	

Figure 4.1: An example of translating from a verb-final language to English. The verb, in **bold**, appears at the end of the sentence, preventing coherent translations until the final source word is revealed.

target language (the language you're translating **to**) comes late in the source language (the language you're translating **from**). A common example is when translating from a verb-final (SOV) language (e.g., German or Japanese) to a verb-medial (SVO) language, (e.g., English). In the example in Figure 4.1, for instance, the main verb of the sentence (in **bold**) appears at the end of the German sentence. An offline (or "batch") translation system waits until the end of the sentence before translating anything. While this is a reasonable approach, it has obvious limitations. Real-time, interactive scenarios—such as online multilingual video conferences or diplomatic meetings—require comprehensible partial interpretations **before** a sentence ends. Thus, a significant goal in interpretation is to make the translation as **expeditious** as possible.

First, we provide some relevant background on simultaneous interpretation, in Section 4.2. We then present three components in our SOV-to-SVO simultaneous MT system: a reinforcement learning framework that uses predictions to create expeditious translations (Section 4.3), a system to predict how a sentence will end (e.g., predicting the main verb; Section 4.5), and a metric that balances quality and expeditiousness (Section 4.4). We combine these components in a framework that learns *when* to begin translating sections of a sentence (Section 4.7).

Section 4.8 combines this framework with a translation system that produces simultaneous translations. We show that our data-driven system can successfully predict unseen parts of the sentence, learn when to trust them, and outperform strong baselines (Section 4.9).

While some prior research has approached the problem of simultaneous translation—we review these systems in more detail in Section 4.10—no current model learns **when** to definitively begin translating chunks of an incomplete sentence. Finally, in Section 4.11, we discuss the limitations of our system: it only uses the most frequent source language verbs, it only applies to sentences with a single main verb, and it uses an idealized translation system. However, these limitations are not insurmountable; we describe how a more robust system can be assembled from these components in Chapter 5.

4.2 Simultaneous Interpretation Background

In this section, we review research in simultaneous interpretation by humans, focusing in particular on the case of SOV-SVO interpretation.

Simultaneous interpretation is the process of, in real-time, translating from a source language to a target language. In this research, we focus on subject-object-verb (SOV) to subject-verb-object (SVO) simultaneous interpretation, which presents unique challenges for humans.

Takagi et al. (2002) performed a corpus-based analysis of Japanese-English and English-Japanese simultaneous interpretation and examined the strategies used by interpreters. For Japanese-English interpretation, they note several general strategies, but one particularly relevant one is the prediction of upcoming words by referencing presentation slides available to the interpreter. Tohyama and Matsubara (2006) perform a similar analysis on the same corpus, but only for English-Japanese, analyzing more fine-grained strategies. Similar strategies were incorporated into He et al. (2015) for creating more monotonic simultaneous machine translations. This was followed up by a computational analysis which demonstrates that the strategies of simultaneous interpreters vis-à-vis those of translators are unique enough to be discriminated by a statistical classifier. (He et al., 2016).

Jörg (1997) measured verb anticipation in a German-English simultaneous interpretation setting of a prepared speech, finding that, 48% of the time, interpreters had no anticipation, 30% of the time they predicted exactly, and 20% of the time they predicted a more general verb that they could use. Only 2% of the time (7/312 opportunities) did they observe incorrect predictions, indicating, they argue, that the interpreters only use verb predictions when they are sure that they are correct (the strategy that we attempts to approximate with reinforcement learning). They also find that professionals predict the exact verb 10% more frequently than students

Bevilacqua (2009) attempted to isolate strategies of simultaneous interpreters from SOV German sentences to AVO Italian sentences. They do this by examining SOV German sentences in the transcripts. Noting that these strategies are generally unconscious, they manually attempt to isolate the strategies of anticipation, ear-voice span management, reformulation, and omission.

Anticipation occurs when the interpreter relies on linguistic or extra-linguistic information—such as phrase probabilities or domain knowledge, respectively—to translate ahead of the speaker in the source language. **Ear-voice span** (EVS) is the delay that occurs between the time when the interpreter hears and the time that the interpreter translates. This was further subcategorized into "**waiting** in silence" and **stalling**. The latter refers to practices such as inserting filler or repeating already-known information to while determining how to deal with new information. **Reformulation** is a strategy employed by the interpreter to produce output that allows for a smooth translation, contingent on the structure of the input. **Compression** is the simplification of the target text to cope with the time and cognitive constraints of simultaneous interpretation. **Omission** refers to elements in the source text which do not appear in the target text.

Goldman-Eisler (1972) posited that the longer ear-voice span for German-English interpretation when compared to English French-English and English-French interpretation is due to the SOV word order. The minimal translatable unit, if the interpreter is to wait for the verb, must be longer in this case (Christoffels, 2004).

In the research presented here, we do not attempt to emulate compression or stalling. We instead focus on minimizing delay (as a proxy for ear-voice span) and employing anticipation, while learning how to judge when to use each.

Human interpreters learn strategies for their profession with experience and practice. As words in the source language are observed, a translator—human or machine—must decide whether and how to translate, while, for certain language pairs, simultaneously predicting future words. We would like our system to do the same. To this end, we model simultaneous MT as a Markov decision process (MDP) and use reinforcement learning to effectively combine predicting, waiting, and translating into a coherent strategy.

4.2.1 States: What is, what is to come

The **state** s_t represents the current view of the world given that we have seen t words of a source language sentence.[2] The state contains information both about what is known and what is predicted based on what is known. To compare the system to a human translator in a decision-making process, the state is akin to

[2] We use t to evoke a discrete version of time. We only allow actions after observing a complete source word.

the translator's cognitive state. At any given time, we have *knowledge* (observations) and *beliefs* (predictions) with varying degrees of certainty: that is, the state contains the revealed words $x_{1:t}$ of a sentence; the state also contains predictions about the remainder of the sentence: we predict the next word in the sentence and the final verb.

More formally, we have a prediction at time t of the next source language word that will appear, $n_{t+1}^{(t)}$, and for the final verb, $v^{(t)}$. For example, given the partial observation "`ich bin mit dem`", the state might contain a prediction that the next word, $n_{t+1}^{(t)}$, will be "`Zug`" and that the final verb $v^{(t)}$ will be "`gefahren`".

We discuss the mechanics of next-word and verb prediction further in Section 4.5; for now, consider these black boxes which, after observing every new source word x_t, make predictions of future words in the source language. This representation of the state allows for a richer set of actions, described below, permitting simultaneous translations that outpace the source language input[3] by predicting the future.

4.2.2 Actions: What our system can do

Given observed and hypothesized input, our simultaneous translation system must decide when to translate them. This is expressed in the form of four actions: our system can **commit** to a partial translation, predict the **next word** and use it to update the translation, predict the **verb** and use it to update the translation, or **wait** for more words.

We discuss each of these actions in turn before describing how they come together to incrementally translate an entire sentence:

Wait Waiting is the simplest action. It produces no output and allows the system to receive more input, biding its time, so that when it does choose to translate, the translation is based on more information.

Commit Committing produces translation output: given the observed source sentence, produce the best translation possible.

Next Word The **next word** action takes a prediction of the next source word and produces an updated translation based on that prediction, i.e., appending the predicted word to the source sentence and

[3] Throughout, "input" refers to source language input, and "output" refers to target language translation.

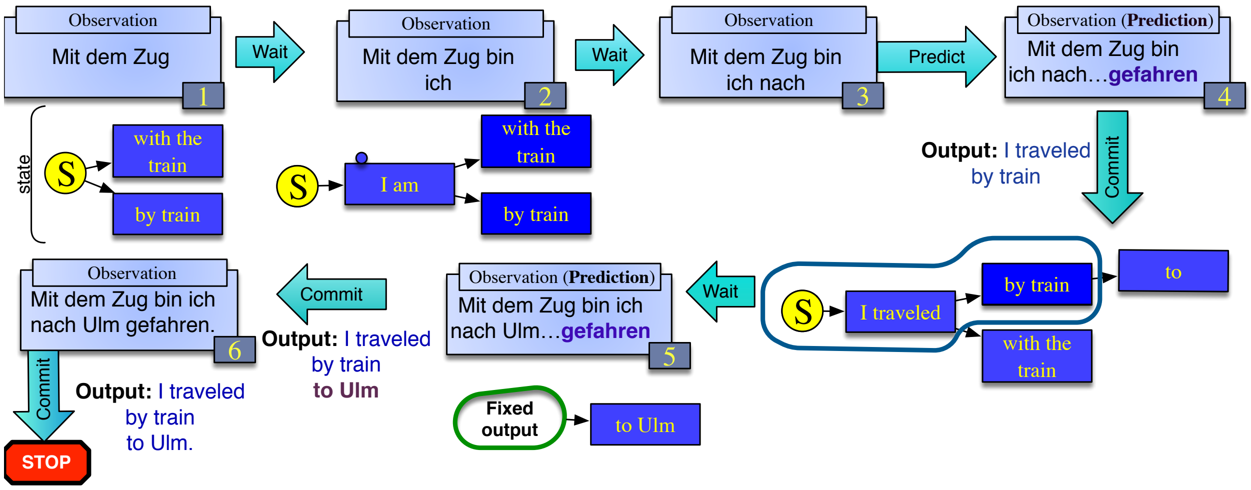


Figure 4.2: A simultaneous translation from source (German) to target (English). The agent chooses to wait until after (3). At this point, it is sufficiently confident to predict the final verb of the sentence (4). Given this additional information, it can now begin translating the sentence into English, constraining future translations (5). As the rest of the sentence is revealed, the system can translate the remainder of the sentence.

translating the new sentence.

Verb Our system can also predict the source sentence's final verb (the last word in the sentence). When our system takes the **verb** action, it uses its verb prediction to update the translation using the prediction, by placing it at the end of the source sentence.

We can recreate a traditional batch translation system (interpreted temporally) by a sequence of **wait** actions until all input is observed, followed by a **commit** to the complete translation. Our system can **commit** to partial translations if it is confident, but producing a good translation early in the sentence often depends on missing information.

4.3 Decision Process for Simultaneous Translation

Having described the state, its components, and the possible actions at a state, we present the process in its entirety. In Figure 4.2, after each German word is received, the system arrives at a new **state**, which consists of the source input, target translation so far, and predictions of the unseen words. The translation system must then take an **action** given information about the current state. The action will result in receiving and translating more source words, transitioning the system to the next state. In the example, for the first few source-language words, the translator lacks the confidence to produce any output due to insufficient information at the state. However, after State 3, the state shows high confidence in the predicted verb "gefahren". Combined with the German input it has observed, the system is sufficiently confident to act on that prediction to produce English translation.

4.3.1 Consensus Translations

Three straightforward actions—**commit**, **next word**, and **verb**—all produce translations. These rely black box access to a translation (discussed in detail in Section 4.8): that is, given a source language sentence fragment, the translation system produces a target language sentence fragment.

Because these actions can happen more than once in a sentence, we must form a single "consensus" translation from all of the translations that we might have seen. If we have only one translation or if translations are identical, forming the consensus translation is trivial, since the translation at unchanged.

Any time our system chooses an action that produces output, the observed input (plus extra predictions in the case of **next-word** or **verb**), is passed into the translation system. That system then produces a complete translation of its input fragment.

Any new words—i.e., words whose target index is greater than the length of any previous translation—are appended to the previous translation.[4] Table 4.1 shows an example of forming these consensus translations.

The consensus translation is formed by allowed a slot for each translated word. Once a slot has been filled, it is set and cannot be changed. Future queries to the translation system can only append words to what has already been translated. Even if future queries to the translation place different words in the earlier positions, the consensus translation ignores these changes, as rewrites are disallowed.

Now that we have defined how states evolve based on our system's actions, we need to know how to select which actions to take. Eventually, we will formalize this as a learned policy (Section 4.7) that maps from states to actions. First, however, we need to define a reward that measures how "good" an action is.

[4] Using constrained decoding to enforce consistent translation prefixes would complicate our method but is an appealing extension.

Pos	Input	Intermediate	Consensus
1			
2	Er	He$_1$	He$_1$
3	Er wurde **gestaltet**	It$_1$ was$_2$ designed$_3$	He$_1$ was$_2$ designed$_3$
4		It$_1$ was$_2$ designed$_3$	He$_1$ was$_2$ designed$_3$
5	Er wurde gestern renoviert	It$_1$ was$_2$ renovated$_3$ yesterday$_4$	He$_1$ was$_2$ designed$_3$ yesterday$_4$

Table 4.1: How intermediate translations are combined into a consensus translation. Incorrect translations (e.g., "he" for an inanimate object in position 3) and incorrect predictions (e.g., incorrectly predicting the verb **gestaltet** in position 5) are kept in the consensus translation. When no translation is made, the consensus translation remains static.

4.4 Objective: What is a good simultaneous translation?

Good simultaneous translations must optimize two objectives that are often at odds, i.e., producing translations that are, in the end, accurate, and producing them in pieces that are presented expeditiously. While there are existing automated metrics for assessing translation quality (Papineni et al., 2002a; Banerjee and Lavie, 2005; Snover et al., 2006), these must be modified to find the necessary compromise between translation quality and expeditiousness. That is, a good metric for simultaneous translation must achieve a balance between translating chunks early and translating accurately. All else being equal, maximizing either goal in isolation is trivial: for accurate translations, use a **batch** system and wait until the sentence is complete, translating it all at once; for a maximally expeditious translation, create **monotone** translations, translating each word as it appears, as in Tillmann et al. (1997) and Pytlik and Yarowsky (2006). The former is not simultaneous at all; the latter is mere word-for-word replacement and results in awkward, often unintelligible translations of distant language pairs.

Once we have predictions, we have an expanded array of possibilities, however. On one extreme, we can imagine a **psychic** translator—one that can completely translate an imagined sentence after one word is uttered—as an unobtainable system. On the other extreme is a standard **batch** translator, which waits until it has access to the utterer's complete sentence before translating anything.

Again, we argue that a system can improve on this by **predicting** unseen parts of the sentence to find a

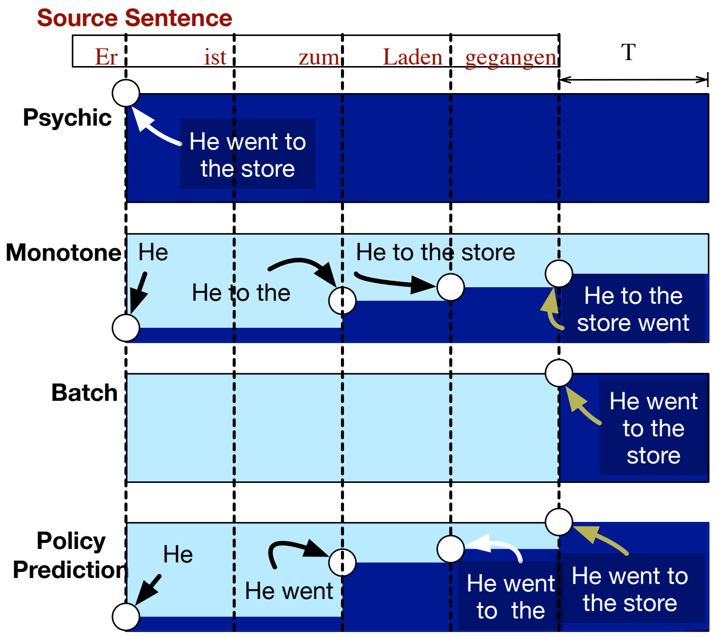


Figure 4.3: Comparison of LBLEU (the area under the curve given by Equation 4.1) for an impossible psychic system, a traditional batch (consecutive) system, a monotone (German word order) system, and our prediction-based system. By correctly predicting the verb "gegangen" (to go), we achieve a better overall translation more quickly.

better tradeoff between these conflicting goals. However, to evaluate and optimize such a system, we must measure where a system falls on the continuum of accuracy versus expeditiousness.

Consider partial translations in a two-dimensional space, with time (quantized by the number of source words seen) increasing from left to right on the x axis and the BLEU score (including brevity penalty against the reference length) on the y axis. At each point in time, the system may add to the consensus translation, changing the precision (Figure 4.3). Like an ROC curve, a good system will be high and to the left, optimizing the area under the curve: the ideal system would produce points as high as possible immediately. A translation which is, in the end, accurate, but which is less expeditious, would accrue its score more slowly but outperform a similarly expeditious system which nevertheless translates poorly.

An idealized psychic system achieves this, claiming all of the area under the curve, as it would have a perfect translation instantly, having no need of even waiting for future input.[5] A batch system has only a

[5] One could reasonably argue that this is not ideal: a fluid conversation requires the prosody and timing between source and target to match exactly. Thus, a psychic system would provide too much information too quickly, making information exchange

narrow (but tall) sliver to the right, since it translates nothing until all of the words are observed.

Formally, let Q be the score function for a partial translation, $\boldsymbol{x}$ the sequentially revealed source words $x_1, x_2, \ldots, x_T$ from time step 1 to T, and $\boldsymbol{y}$ the partial translations $y_1, y_2, \ldots, y_T$, where T is the length of the source language input. Each incremental translation y_t has a BLEU-n score with respect to a reference $\boldsymbol{r}$. We apply the usual BLEU brevity penalty to all the incremental translations (initially empty) to obtain **latency-BLEU** (LBLEU),

$$Q(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{T} \sum_t \text{BLEU}(y_t, r) + T \cdot \text{BLEU}(y_T, r) \tag{4.1}$$

The LBLEU score is a word-by-word integral across the input source sentence. As each source word is observed, the system receives a reward based on the BLEU score of the partial translation. LBLEU, then, represents the sum of these T rewards at each point in the sentence. The score of a simultaneous translation is the sum of the scores of all individual segments that contribute to the overall translation.

We multiply the final BLEU score by T to ensure good final translations in learned systems to compensate for the implicit bias toward low latency.[6]

4.5 Predicting Verbs and Next Words

The **next** and **verb** actions depend on predictions of the sentence's next word and final verb; this section describes our process for predicting verbs and next words given a partial source language sentence.

The prediction of the next word in the source language sentence is modeled with a left-to-right language model. This is (naïvely) analogous to how a human translator might use a mental "language model" to guess upcoming words to gain some speed by completing, for example, collocations before they are uttered. We use a simple Kneser-Ney (Kneser and Ney, 1995) bigram language model for next-word prediction. For verb prediction, we use a generative model that combines the prior probability of a particular verb v, $p(v)$, with the likelihood of the source context at time t given that verb (namely, $p(x_{1:t} \| v)$), as estimated by a smoothed

unnatural. However, we take the information-centric approach: more information faster is better.

[6] One could replace T with a parameter, β, to bias towards different kinds of simultaneous translations. As $\beta \to \infty$, we recover batch translation.

Kneser-Ney language model. The context, $x_{1:t}$, consists of all words observed. We model $p(x_{1:t} \| v)$ with verb-specific n-gram language models. The predicted verb $v^{(t)}$ at time t is then:

$$\arg\max_{v} p(v) \prod_{i=1}^{t} p(x_i \| v, x_{i-n+1:i-1}) \tag{4.2}$$

where $x_{i-n+1:i-1}$ is the $n-1$-gram context. To narrow the search space, we consider only the 100 most frequent final verbs, where a "final verb" is defined as the sentence-final sequence of verbs and particles as detected by a German part-of-speech tagger (Toutanova et al., 2003).[7]

4.6 A Model for Verb-final Sentences

Human interpreters learn strategies for their profession with experience and practice. (Gile, 2009) We would like our system to do the same. Simultaneous machine translation is an inherently stochastic problem. As words are observed, a translator, human or machine, must make decisions based on new information. We are navigating a decision space with incomplete and constantly updated knowledge, and we would like to make the best possible decision at any given state. This is a Markov decision process.

Reinforcement learning (Thrun and Littman, 2000) affords language systems the same opportunity to learn from experience as humans, potentially even allowing them to learn similar strategies. The fundamental components of reinforcement learning systems are (1) a **state** that describes a changing environment; (2) a set of **actions** that an algorithm can use to affect the underlying state; and (3) a **reward** that retrospectively measures how good (or bad) a particular action was in a state. We will discuss each of these in turn.

4.6.1 State

The **state** s represents the current view of the world at a given point t in time: the state contains information both about what is known and what is predicted based on what is known. To compare the system to a human translator in a decision-making process, the state is akin to one's cognitive state. At any given time, we have *knowledge* (observations), and *beliefs* (predictions) of varying degrees of certainty (probabilities).

[7] This has the obvious disadvantage of ignoring morphology and occasionally creating duplicates of common verbs that have may be associated with multiple particles; nevertheless, it provides a straightforward verb to predict.

We consider each position, t, in a sentence to be revealed at a discrete point in time; as t increases, so does the revealed sentence size, in words. That is, t and the number of words revealed are the same value, since each revealed word represents a time step. The state contains the revealed words x_t (which we use as a basic linguistic unit here) of a sentence. This is known. The state also encodes predictions about the remainder of the sentence and associated probabilities. More specifically, we predict the next word in the sentence and the final verb. This is "expected" to the degree that the probabilities indicate. For concreteness, we have a prediction at time t of the next source language words that will appear, $n_{t+1}^{(t)}$, and for the final verb, $v^{(t)}$. For example, given the partial observation "`ich bin mit dem`", the state might contain a prediction that the next word, $n_{t+1}^{(t)}$, will be "`Zug`" and that the verb $v^{(t)}$ will be "`gefahren`". The state s_t is represented by a feature vector $\phi(x_t, n_{t+1}^{(t)}, v^{(t)})$, where ϕ is the feature map (Section 4.7).

The predictions are provided by n-gram language models. We create a verb-specific language model for the most frequent source language verbs and create a n-gram prediction as the product of n-gram contexts c in the source input x_t at time t.

$$v^{(t)} \equiv \arg\max_{v} \prod_{c \in x_t} p(c \,|\, v) p(v). \tag{4.3}$$

Similarly, we predict the next word in the source language; it is the most likely completion of the last observed source context. We can easily imagine a human translator using such guesses to try to gain some speed by finishing, for example, collocations before they are uttered.

4.6.2 Actions

An **action** a specifies the decisions our system has to make to produce translations given information from the state s. In order to produce translated output as its input arrives, the system must decide when to wait for further input and when to commit to partial translations. An example for this process is shown in Figure 4.2 using the representation of intermediate translation hypotheses of Koehn (Koehn, 2009).

At every point in time, based on the **state**, the system must take an action . There are four possible actions our translation agent can take after it has seen source input $x_1 \ldots x_t$: it can **commit** to a partial translation, apply the current prediction of the **next word**, apply the current predicton of the **verb**, or **wait** for more words. We describe each of these in turn.

Each of these actions except for **wait** sends the current observations to the translator, producing an intermediate translation which must be appended to the previous partial translation to form the updated translation. Each token in an intermediate translation is associated with a target word position. The first intermediate translation to be placed in a target word position—i.e., the intermediate translation made with the lowest source index t—will be the word used in the updated consensus translation. This is demonstrated in Table 4.1. Thus, we do not allow corrections or restarts.

commit

If the agent commits, it receives a translation $\tau(x_{1:t})$ from the omniscient translator. The output of this is used as the intermediate translation and added to the consensus translation.

next word

If the agent takes the next word prediction, then it gets a translation $\tau(x_{1:t}n_t)$, appending the next word prediction n_t to the current observed source string. It gets the translation for that string as the intermediate translation and it is added to the consensus translation.

verb

If the agent takes the verb action, it gets the verb prediction and receives a translation $\tau(x_{1:t} \ldots v_t)$, where v_t is the current verb prediction. This translation is the intermediate translation and is added to the consensus.

wait

If the agent waits, then the translation remains unchanged from the previous point in time, and the consensus translation remains unchanged.

4.7 Learning a Policy

We have a framework (states and actions) for simultaneous machine translation and a metric for assessing simultaneous translations. We now describe the use of reinforcement learning to learn a **policy**, a mapping from states to actions, to maximize LBLEU reward.

We use imitation learning (Abbeel and Ng, 2004; Syed et al., 2008): given an optimal sequence of actions, we learn a generalized policy that maps states to actions. This can be viewed as classification (Langford and Zadrozny, 2005): a state is represented as a feature vector, the loss corresponds to the quality of the action, and the output of the classifier is the action that should be taken in that state.

In this section, we explain each of these components: generating an optimal policy, representing states through features, and learning a policy that can generalize to new sentences.

4.7.1 Optimal Policies

Because we will eventually learn policies via a classifier, we must provide training examples to our classifier. These training examples come from an **oracle policy** π^* that demonstrates the optimal sequence—i.e., with maximal LBLEU score—of actions for each sequence. Using dynamic programming, we can determine such actions for a fixed translation model.[8] From this oracle policy, we generate training examples for a supervised classifier. State s_t is represented as a tuple of the observed words $x_{1:t}$, predicted verb $v^{(t)}$, and the predicted word $n_{t+1}^{(t)}$. We represent the state to a classifier as a feature vector $\phi(x_{1:t}, n_{t+1}^{(t)}, v^{(t)})$.

4.7.2 Feature Representation

We want a feature representation that will allow a classifier to generalize beyond the specific examples on which it is trained. We use several general classes of features: features that describe the input, features that describe the possible translations, and features that describe the quality of the predictions.

Input We include both a bag of words representation of the input sentence as well as the most recent word and bigram to model word-specific effects. We also use a feature that encodes the length of the source sentence.

Prediction We include the identity of the predicted verb and next word as well as their respective probabilities under the language models that generate the predictions. If the model is confident in the prediction, the classifier can learn to more so trust the predictions.

[8] This is possible for the limited class of consensus translation schemes discussed in Section 4.3.1.

Translation In addition to the state, we include features derived from the possible actions the system might take. This includes a bag of words representation of the target sentence, the score of the translation (decreasing the score is undesirable), the score of the current consensus translation, and the difference between the current and potential translation scores.

4.7.3 Policy Learning

Our goal is to learn a classifier that can accurately mimic the oracle's choices on previously unseen data. However, at test time, when we run the learned policy classifier, the learned policy's state distribution may deviate from the optimal policy's state distribution due to imperfect imitation, arriving in states not on the oracle's path. To address this, we use SEARN (Daumé et al., 2009), an iterative imitation learning algorithm that allows us to train our model from optimum examples. We learn from the optimal policy in the first iteration, as in standard supervised learning; in the following iterations, we run an interpolated policy

$$\pi_{k+1} = \epsilon\pi_k + (1-\epsilon)\pi^*, \tag{4.4}$$

with k as the iteration number and ϵ the mixing probability. We collect examples by asking the policy to label states on its path. The interpolated policy will execute the optimal action with probability $1-\epsilon$ and the learned policy's action with probability ϵ. In the first iteration, we have $\pi_0 = \pi^*$.

Mixing in the learned policy allows the learned policy to slowly change from the oracle policy. As it trains on these no-longer-perfect state trajectories, the state distribution at test time will be more consistent with the states used in training.

SEARN learns the policy by training a cost-sensitive classifier. Besides providing the optimal action, the oracle must also assign a cost to an action

$$\mathcal{C}(a_t, \boldsymbol{x}) \equiv Q(\boldsymbol{x}, \pi^*(x_t)) - Q(\boldsymbol{x}, a_t(x_t)), \tag{4.5}$$

where $a_t(x_t)$ represents the translation outcome of taking action a_t. The cost is the regret of not taking the optimal action.

4.8 Translation System

The focus of this work is to show that given an effective batch translation system and predictions, we can learn a policy that will turn this into a simultaneous translation system. Thus, to separate translation errors from policy errors, we perform experiments with a nearly optimal translation system we call an **omniscient** translator.

More realistic translation systems will naturally lower the objective function, often in ways that make it difficult to show that we can effectively predict the verbs in verb-final source languages. For instance, German to English translation systems often drop the verb; thus, predicting a verb that will be ignored by the translation system will not be effective.

The omniscient translator translates a source sentence correctly once it has been fed the appropriate source words as input. There are two edge cases: empty input yields an empty output, while a complete, correct source sentence returns the correct, complete translation. Intermediate cases—where the input is either incomplete or incorrect—require using an alignment. The omniscient translator assumes as input a reference translation r, a partial source language input $x_{1:t}$ and a corresponding partial output y. In addition, the omniscient translator assumes access to an **alignment** between r and x. In practice, we use the HMM aligner (Vogel et al., 1996b; Och and Ney, 2003).

We first consider incomplete but correct inputs. Let $y = \tau(x_{1:t})$ be the translator's output given a partial source input $x_{1:t}$ with translation y. Then, $\tau(x_{1:t})$ produces all target words y_j if there is a source word x_i in the input aligned to those words—i.e., $(i, j) \in a_{x,y}$—*and* all preceding target words can be translated. (That translations must be contiguous is a natural requirement for human recipients of translations). In the case where y_j is unaligned, the closest aligned target word to y_j that has a corresponding alignment entry is aligned to x_i; then, if x_i is present in the input, y_j appears in the output. Thus, our omniscient translation system will always produce the correct output given the correct input.

However, our learned policy can make wrong predictions, which can produce partial translations y that do **not** match the reference. In this event, an incorrect source word $\tilde{x}_i$ produces incorrect target words $\tilde{y}_j$, for all $j\colon (i, j) \in a_{x,y}$. These $\tilde{y}_j$ are sampled from the IBM Model 1 lexical probability table multiplied by the source language model $\tilde{y}_j \sim \text{Mult}(\theta_{\tilde{x}_i}) p_{LM}(\tilde{\boldsymbol{x}})$.[9] Thus, even if we predict the correct verb using a **next word** action, it will be in the wrong position and thus generate a translation from the lexical probabilities. Since translations based on Model 1 probabilities are generally inaccurate, the omniscient translator will do very well when given correct input but will produce very poor translations otherwise.

4.9 Experiments

In this section, we describe our experimental framework and results from our experiments. From aligned data, we derive an omniscient translator. We use monolingual data in the source language to train the verb predictor and the next word predictor. From these features, we compute an optimal policy from which we train a learned policy.

4.9.1 Data sets

For translation model and policy training, we use data from the German-English Parallel "de-news" corpus of radio broadcast news (Koehn, 2000), which we lower-cased and stripped of punctuation. A total of $48,601$ sentence pairs are randomly selected for building our system. Of these, we use 70% ($34,528$ pairs) for training word alignments.

For training the translation policy, we restrict ourselves to sentences that end with one of the 100 most frequent verbs (see Section 4.5). This results in a data set of 4401 training sentences and 1832 test sentences from the de-news data. We did this to narrow the search space (from thousands of possible, but mostly very infrequent, verbs).

We used 1 million words of news text from the Leipzig Wortschatz (Quasthoff et al., 2006) German corpus to train 5-gram language models to predict a verb from the 100 most frequent verbs.

For next-word prediction, we use the $18,345$ most frequent German bigrams from the training set to provide a set of candidates in a language model trained on the same set. We use frequent bigrams to reduce the computational cost of finding the completion probability of the next word.

[9] If a policy chooses an incorrect unaligned word, it has no effect on the output. Alignments are position-specific, so "wrong" refers to position and type.

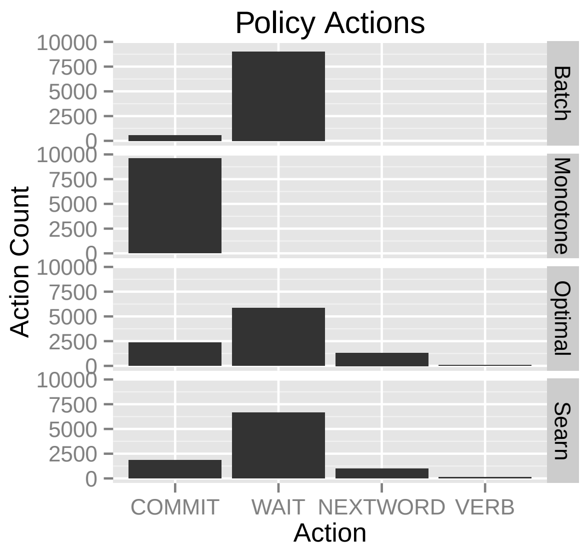


Figure 4.4: Histogram of actions taken by the policies.

4.9.2 Training Policies

In each iteration of SEARN, we learn a multi-class classifier to implement the policy. The specific learning algorithm we use is AROW (Crammer et al., 2013), though others can be substituted. In the complete version of SEARN, the cost of each action is calculated as the highest expected reward starting at the current state minus the actual roll-out reward. However, computing the full roll-out reward is computationally very expensive. We thus use a surrogate binary cost: if the predicted action is the same as the optimal action, the cost is 0; otherwise, the cost is 1. We then run SEARN for five iterations. Results on the development data indicate that continuing for more iterations yields no benefit.

4.9.3 Policy Rewards on Test Set

In Figure 4.5, we show performance of the optimal policy *vis-à-vis* the learned policy, as well as the two baseline policies: the batch policy and the monotone policy. The x-axis is the percentage of the source sentence seen by the model, and the y-axis is a smoothed average of the reward as a function of the percentage of the sentence revealed. The monotone policy's performance is close to the optimal policy for the first half of the sentence, as German and English have similar word order, though they diverge toward the end. Our

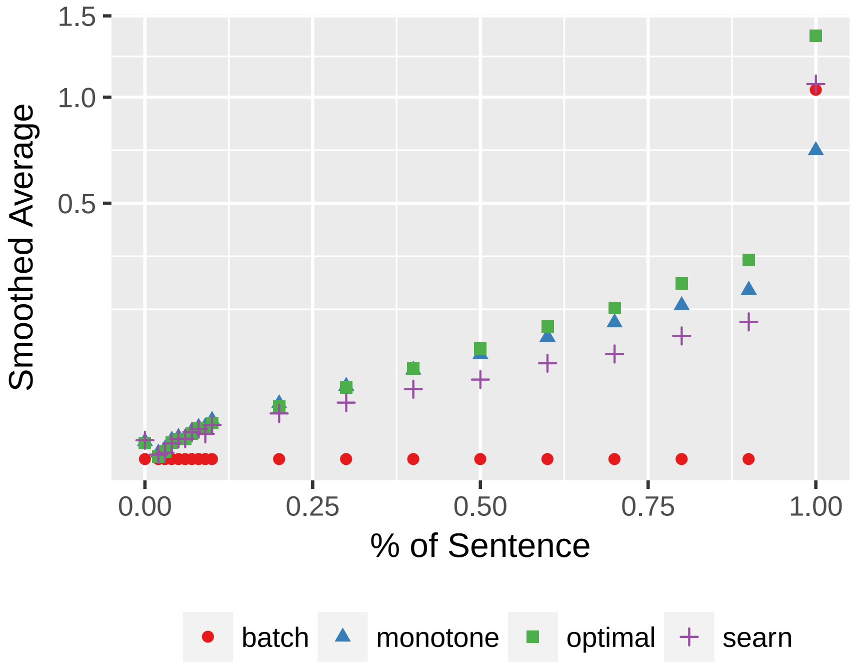


Figure 4.5: The final reward of policies on German data. Our policy outperforms all baselines by the end of the sentence.

learned policy outperforms the monotone policy toward the end and of course outperforms the batch policy throughout the sentence.

Figure 4.4 shows the distribution of actions taken by each policy. The batch policy always commits at the end, and the monotone policy commits at each position. Our learned policy has an action distribution similar to that of the optimal policy. Due to the inaccurate verb predictions, the optimal policy rarely uses the verbs. The SEARN policy is slightly more conservative, but the distribution of actions of the test data is, as we would like, similar to that of the optimal policy: it learns to generalize when to use take each of the four actions.

4.9.4 What Policies Do

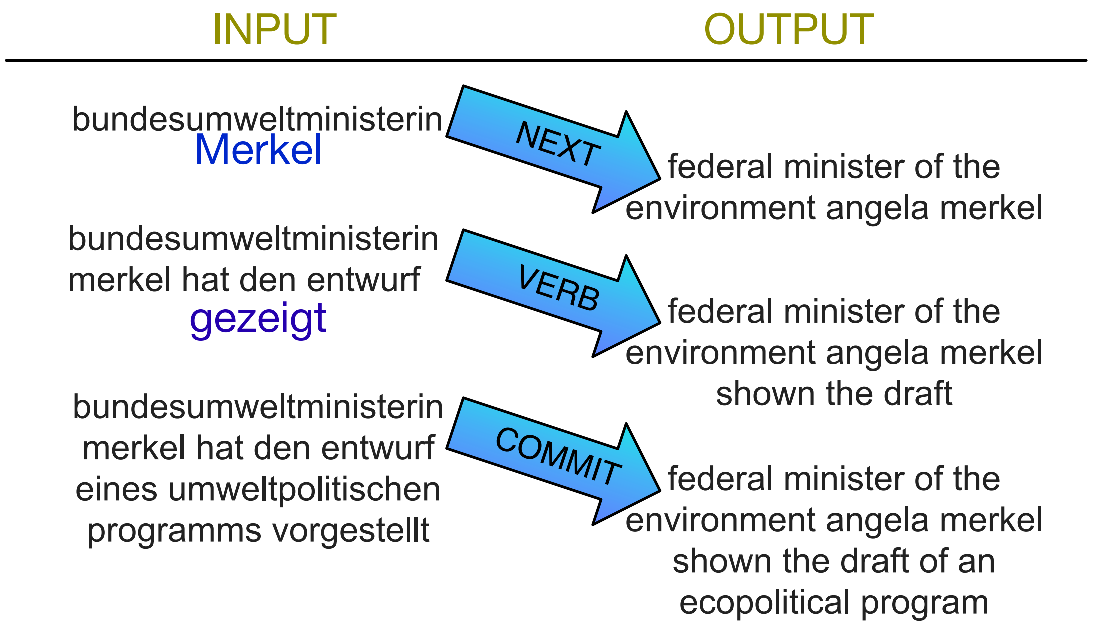

INPUT
OUTPUT
bundesumweltministerin Merkel
NEXT
federal minister of the environment angela merkel
bundesumweltministerin merkel hat den entwurf gezeigt
VERB
federal minister of the environment angela merkel shown the draft
bundesumweltministerin merkel hat den entwurf eines umweltpolitischen programms vorgestellt
COMMIT
federal minister of the environment angela merkel shown the draft of an ecopolitical program

Figure 4.6: An imperfect execution of a learned policy. Despite choosing the wrong verb “gezeigt” (showed) instead of “vorgestellt” (presented), the translation retains the meaning.

Figure 4.6 shows a policy that, predicting incorrectly, still produces sensible output. The policy correctly intuits that the person discussed is Angela Merkel, who was the environmental minister at the time, but the policy uses an incorrectly predicted verb. Because of our poor translation model (Section 4.8), it renders this word as “shown”, which is a poor translation. However, it is still comprehensible, and the overall policy is similar to what a human would do: intuit the subject of the sentence from early clues and use a more general verb to stand in for a more specific one.

4.10 Relationship to Previous Work

Just as MT was revolutionized by statistical learning, we suspect that simultaneous MT will similarly benefit from this paradigm, both from a systematic system for simultaneous translation and from a framework for learning how to incorporate predictions.

Simultaneous translation has been dominated by rule and parse-based approaches (Mima et al., 1998a; Ryu et al., 2006). In contrast, although Verbmobil (Wahlster, 2000) performs incremental translation using a statistical MT module, its incremental decision-making module is rule-based. Other recent approaches in speech-based systems focus on waiting until a pause to translate (Sakamoto et al., 2013) or using word alignments (Ryu et al., 2012) between languages to determine optimal translation units.

Unlike our work, which focuses on prediction and learning, previous strategies for dealing with SOV-to-SVO translation use rule-based methods (Mima et al., 1998b) (for instance, passivization) to buy time for the translator to hear more information in a spoken context—or use phrase table and reordering probabilities to decide where to translate with less delay (Fujita et al., 2013a). Oda et al. (2014) is the most similar to our work on the translation side. They frame word segmentation as an optimization problem, using a greedy search and dynamic programming to find segmentation strategies that maximize an evaluation measure. However, unlike our work, the direction of translation was from *from* SVO to SVO, obviating the need for verb prediction. Simultaneous translation is more straightforward for languages with compatible word orders, such as English and Spanish (Fügen, 2008). Similarly, in He et al. (2015) we examine the role of reordering the training data in a translation system to reduce delay in a translation system, and He et al. (2016) examine the different kinds of language used by human translators and interpreters and demonstrate the ability to classifier the text.

To our knowledge, the only attempt to specifically predict verbs or any late-occurring terms (Matsubara et al., 2000a) uses pattern matching on what would today be considered a small data set to predict English verbs for Japanese to English simultaneous MT.

Simultaneous translation is connected to the psycholinguistic task of “shadowing” (Miller and Isard, 1963) of asking one person to repeat what another says as closely as possible, a task that like translation requires repairs and prediction (Marslen-Wilson, 1973). Our A Incorporating verb predictions into the translation process is a significant component of our framework, though n-gram models strongly prefer highly frequent verbs. The psycholinguistics literature has also given attention to the problem of verb prediction from a human sentence processing perspective. (Bock and Levelt, 2002; Ferreira, 2000; Kempen and Huijbers, 1983) maintain that verbs have a privileged place in sentence planning. Verb prediction might be improved by applying the insights from psycholinguistics. Ferreira (2000) argues that verb lemmas are required early in sentence production—prior to the first noun phrase argument—and that multiple possible syntactic hypotheses are maintained in parallel as the sentence is produced. Schriefers et al. (1998) argues that, in simple German sentences, non-initial verbs do not need lemma planning at all. Momma et al. (2015), investigating these prior claims, argues that the abstract relationship between the internal arguments and verbs triggers **selective** verb planning. Other investigations include work on predictive strategies of humans in sentence processing (Kamide et al., 2003; Van Berkum et al., 2005; Lau et al., 2009) and contextual probability updating (Jurafsky, 1996; Hale, 2003; Knill and Pouget, 2004) that can manifest themselves both behaviorally and neurologically (Kiani and Shadlen, 2009; Frank et al., 2013). We cover this in more detail in Section 3.1.

4.11 Conclusion

We have shown that, when we have an ideal translation system, reinforcement learning with linguistic predictions, even extremely unreliable, simplistic predictions, can improve SOV-SVO simultaneous machine translation in German by optimizing to an automated metric.

Having shown that the learning problem is feasible under simplified circumstances that provide some innoculation the messy artifacts of using a real translation system, the natrual next step is to integrate our improved verb predictions from Chapter 3 into this framework and to create a system closer to one that we can actually use. We do this in Chapter 5: we perform similar experiments with a real translation system, and we integrate the improved verb predictions from Chapter 3.

Chapter 5

Reinforcement Learning for Simultaneous Machine Translation with a Phrase-based Decoder and Discriminative Verb Predictions

Our work thus far on simultaneous machine translation has shown that, with appropriate features, reinforcement learning can be used to perform simultaneous machine translation under simplifying assumptions: predictions, combined with a reinforcement learning framework, improve the performance of simultaneous machine translation. There are, however, several limitations to this system: first, while useful for isolating the learning problem from the translation problem, we have only shown our approach to be effective with an unrealistic, omniscient translation system (Section 4.8), that is not replicable in a real-world setting. While this allows us to abstract away the learning problem from the decoding problem, it does not show the effectiveness of the system in a with a real translation system. Second, the n-gram-based verb prediction models are highly inaccurate (Figure 3.9a), and our analysis shows that the system learns to ignore largely them, as it should, since the optimal policy demonstrates that they usually hurt performance if used. (Figure 4.4) In this chapter, we show that our approach is effective even with the added complexity of a real translation system. We also replace the language model-based verb predictor with the discriminative model from Chapter 3 and use a full language model rather than one that only uses frequent bigrams. Our goal in this chapter is not to describe the details of a system one would actually *deploy*—we have not yet achieved this—but rather to demonstrate that our learning framework is adaptable to the messiness of a real translation system, and, secondarily, to demonstrate that the primary components (discriminative verb prediction, language model-based word prediction, and reinforcement learning-based decision-making) are suitable for such a system.

5.1 System and Model Improvements

5.1.1 Phrase-based Translation System

Our original implementation, which used an omniscient translator (Section 4.8), demonstrates that reinforcement learning is a viable technique for learning a model for machine learning-based simultaneous machine translation. The omniscient translator allows us to abstract away from translation issues, in favor of focusing on learning issues.

There are, however, several compromises in such a system. Chief among these is that the omniscient translator is infeasible in real-world settings. We therefore replace the omniscient translation system with a standard statistical translation system. We use Joshua (Post et al., 2015), a state-of-the-art, open source phrase-based translation system as a drop-in replacement for the omniscient translator that is reasonably fast and representative of current phrase-based systems. We use Joshua's hierarchical phrase-based grammar (Chiang, 2007).

We opt to use the same slot-based method as prior experiments (Section 4.3.1, Table 4.1) for combining the the translations: rewrites are disallowed, which allows us to both use dynamic programming to find optimal policies in a feasible amount of time and to calculate the cumulative reward straightforwardly at test time.

5.1.2 Next-word Prediction Expansion and Speedup

We also improve the method of next-word prediction. In our prior experiments (Chapter 4), we modeled only the most frequent bigrams, and, at test time, calculated, one-by-one, the continuation probabilities of each possible next word, selecting the most probable. This is an extremely slow approach which omits a large portion of the vocabulary due to the computational cost.

Auto-completion algorithms, popular in search engines and source code editors, allow for fast lookup of probably next words. In our improved system, we use the auto-completion framework of Lucene (Białecki et al., 2012), a popular open source information retrieval software package, to create a trigram language model (Brants et al., 2007) trained on one million sentences. This implementation can make predictions very quickly, which is necessary in any real-world scenario. It is also able predict the entire vocabulary of the training data instead of a small subset.

5.1.3 Discriminative Improved Verb Prediction

Our proof-of-concept system in Chapter 4 relied on rudimentary language model-based verb prediction. We have shown that discriminative verb prediction is vastly superior our language model-based approach (Chapter 3). We therefore replace our language model-based verb predictor with our discriminative verb prediction system, increasing the number of correct and useful verb predictions to which the reinforcement learning system has access. Instead of the n-gram probability, we use the probability returned by the one-vs-all logistic regression classifier (Section 3.3) as the score for each prediction, $p_v(v|c)$, i.e., the probability of the verb given the context according to the base classifier trained on verb v vs. all other verbs. A theoretical consequence of this is that $\sum_{v \in V} p_v(v|c) \neq 1$, but we do not require the scores to adhere to the Kolmogorov axioms for our purposes, as long as the relative order is maintained.[1]

5.2 Model Particulars

5.2.1 Translation Model

For the translation model, we use a hierarchical phrase-based (Chiang, 2007) German-English model trained with Joshua (Post et al., 2015) on a diverse set of of parallel corpora.[2]

[1] One-vs-all is a heuristic algorithm for multiclass classification problems. While it is widely used in practice due to its performance, it is known to be mathematically inconsistent (Tewari and Bartlett, 2007; Wang et al., 2010), but few practitioners care because it works so well in practice (Rifkin and Klautau, 2004)

[2] This was created Paul McNamee on 10/18/16 and is available at https://cwiki.apache.org/confluence/display/JOSHUA/Language+Packs. The model was trained on the combined German-English bitext of the following corpora from OPUS[3] [4] (Tiedemann, 2009)

- **DGT Translation Memory** This corpus contains "most, although not all, of the documents which make up the Acquis Communautaire."[5]
- **ECB** (Tiedemann, 2009) Website and documentation of the European Central Bank web site.
- **EMEA** Corpus of European Medicines Agency documents.
- **EUbookshop** (Skadiņš et al., 2014) Corpus extracted EUBookshop, a document repository.
- **Europarl** (Tiedemann, 2012) Corpus sourced from the European Parliament web site.
- **GlobalVoices** (Tiedemann, 2012) A news corpus sourced from the Global Voices[6] web site.
- **GNOME** (Tiedemann, 2009) GNOME localization files.
- **JRC-Acquis** (Steinberger et al., 2006) Collection of legislative text from the 1950s to the present.
- **KDE4** KDE localization files.

5.2.2 Prediction Models

For next word prediction, we use the Lucene suggester API to build an efficient trigram language model Brants et al. (2007). We train on the same data used in our verb prediction experiments in Chapter 3, without limiting ourselves to verb-final sentences or the top k verbs.

For verb prediction, we use the same German model as used in our discriminative verb prediction experiments.

5.3 Experiments

5.3.1 Experimental Setup

We now present the results out our evolved translation system. We again use SEARN (Daumé III et al., 2006) to learn a policy for selecting actions, as described in Section 4.9. We train the SEARN system on 2000 verb-final sentences from de-news corpus (Koehn, 2000), the same corpus as in the omniscient translation experiments (Chapter 4) for four iterations, using AROW (Crammer et al., 2013) as the classifier to learn the policy, and we test on a disjoint set of 2930 verb-final sentences. Unlike in the omniscient translator experiments, we do not limit ourselves to sentences ending in the 100 most frequent verbs. This makes the problem more difficult. Again, we train the language model on 1 million sentences from the Wortschatz (Quasthoff et al., 2006) German corpus, but without limiting ourselves to frequent bigrams at prediction time as in the omniscient experiments. Predictions and translation take place in real time.

- **News-Commentary11** (Tiedemann, 2012), OpenSubtitles2016[7] (Lison and Tiedemann, 2016) Database of TV and movie subtitles.
- **Tanzil** Collection of Qu'ran translations.
- **Tatoeba** Collection of sentences from the Tatoeba project[8] , many of which originate from the Tanaka corpus (Tanaka, 2001)
- **TED2013** TED Talks parallels corpus.
- **Ubuntu** Ubuntu Linux localization files.
- **Wikipedia** Wikipedia translated sentences.

5.3.2 Results

We now discuss the results of our experiments with the Joshua, our language model, and discriminative verb prediction.

In our results (Figure 5.1), we find a similar pattern to our omniscient translation experiments (Figure 4.5), though the overall scores are significantly lower. This is expected, due to the higher uncertainty inherent to using a phrase-based translation system. Moreover, in contrast to the results of our omniscient translator experiments, the learned policy slightly outperforms the monotone policy even earlier in the sentence, as opposed to merely at the end. At no point in the sentence is the average reward below the baselines, but it manages to produce a higher translation overall.

Our results suggest that, as in the omniscient translation experiments, we learn a policy that outperforms the baselines (Figure 5.1), even with the added uncertainty and complexity of a phrase-based translation system not specifically designed for incremental translation: The SEARN policy learns to select the appropriate actions to maximize its score, coping with the limitations of the translation system.

Additionally, with our improved verb prediction and next-word prediction, the policy uses these predictions more often then before (Figure 5.2). Compared to the actions of the omniscient system (Figure 4.4) The optimal policy's increased use of the **next word** and **verb** actions, furthermore, indicates that this is the correct strategy: we see that the distribution of the actions for the learned policy is similar to those of the optimal policy.

5.4 Analysis

In this section, we examine some examples to glean some insight into the behavior of our system. Examination of the translation output of our system reveals that a persistent issue is one of untranslated words, often compound nouns or proper nouns that did not appear in the training data. The latter is to be expected in any system tested on real data. The former is a well-known issue in machine translation from German. (Och et al., 1999; Popović et al., 2006; Yang and Kirchhoff, 2006) Other instances appear to be artifacts of feeding the decoder incomplete or bizarre phrases, leading to undesirable behavior from the decoder.

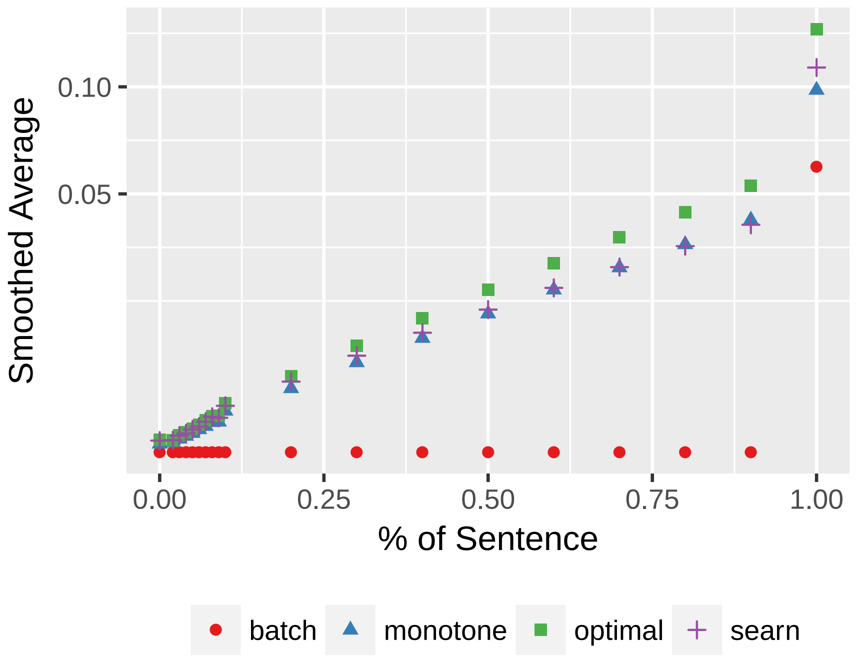


Figure 5.1: Cumulative reward for each the two baseline policies, the optimal policy, and the learned policy. As in the omniscient translator experiments, SEARN outperforms at the end of the sentence, but with the phrase-based translation system, it also outperforms the baselines along the way.

We also find that the translations output of the underlying decoder is not monotonic: adding a word to the German input of the decoder may *shorten* the English output of the decoder. Because our system does not allow backtracking or correction, this does not shorten the output of the simultaneous translation itself. Recall that if we have English translations for word positions 0 to k, words $k + 1$ or greater can be altered. That is, word positions 0 to k are fixed. (Section 4.3.1 describes the algorithm for incremental output). Any decrease in the translation length occurs only at the level of the underlying decoder, which can be viewed as a black box.

Yet another related, but slightly different, problem we observed is that of “stalled sentences”. This occurs when there is no new translation output after a given position, regardless of more observed words. That is, even at the final commit, there is no further incremental English translation. This can happen because the decoder’s output length does not grow; it can also occur if the decoder simply cannot adequately translate its input.

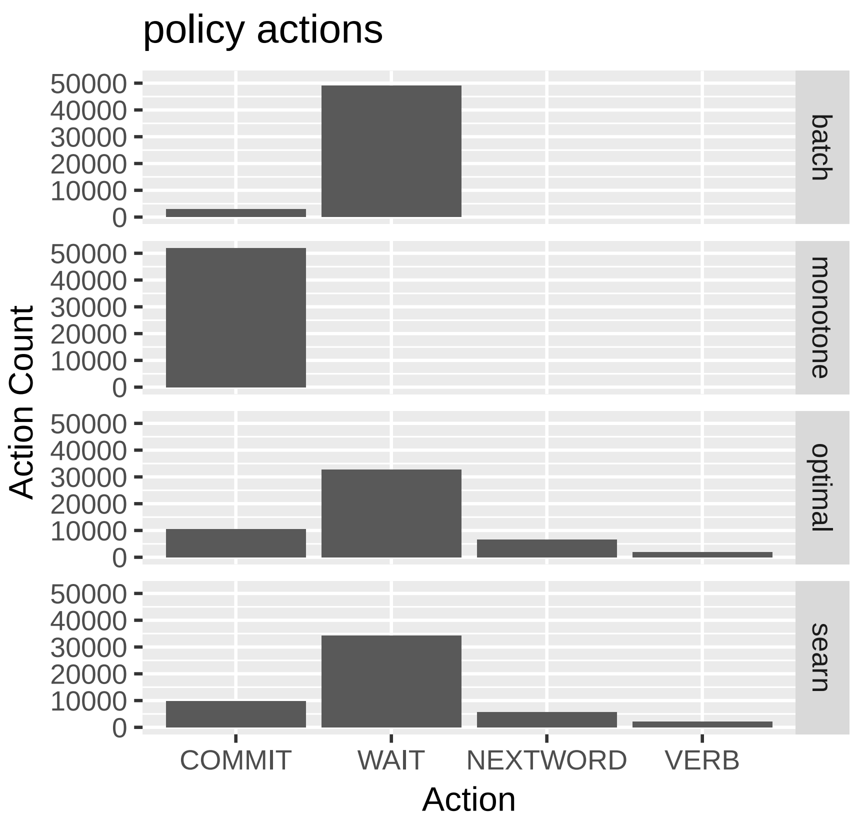


Figure 5.2: Distribution of actions for each policy with phrase-based translation system. The distribution of actions for the optimal and learned policies is similar, and the **verb** and **next word** actions are both (1) more helpful and (2) used more often by the learned policy.

Since the system does not know the length of the output of the decoder, we can at times get strange results. Consider the following undesirable translations:

- **Word/Phrase Repetition** “The main measures called the registrar planned planned tax reform and planned tax. . .” (Searn policy)
- **Stalled Sentence** “Without the agreement of the former soviet president” (Searn policy).

Many sentences have some combination of these problems. The reordering that takes place in the underlying decoder based on the input exacerbates these issues. Despite these shortcomings, however, the system does manage to learn to work with what it has.

5.5 Conclusion and Future Work

We show that our reinforcement learning system can outperform our baselines when using a real translation system. We also show that the policy learns to effectively utilize the improved next-word and verb predictions. While this is only a first small step toward the eventual goal of creating a truly usable simultaneous translation system, we have nevertheless demonstrated that reinforcement learning can learn to make use of what it has at its disposal. As more research improves predictive accuracy in a stochastic setting, we can expect more useful results.

There are still several weaknesses in our approach that must be addressed.

- **Verb prediction** Our verb prediction system is still rather primitive, relying as it does on bags of simple features. Our most linguistically-informed features are case markers, and we do not take into account linguistic insights into how clause identification aids prediction in humans.

- **Translation System** While we have shown our approach to be viable for a real translation system, the translation system we use is not designed for simultaneous machine translation. If our system can learn to translate phrases accurately, this can potentially be worked around, but more research is necessary to determine what kinds of translation systems are most suitable for this framework. More specifically, we do not use explicit incremental decoding, instead relying on the policy to maximize independent queries to the translation system. We also do not take into account the specifics of words or phrases already committed.

- **Action Space** Our action space is still rudimentary. While verb prediction and next-word prediction help our system, there is already work on predicting syntactic constituents to help this task (Oda et al., 2015).

- **Feature Space** Our features are simple and do not take into account potential linguistic insights, such as identifying potential clause boundaries.

- **Using Reordered Corpora** In (He et al., 2015), we use syntax-based reordering to reduce lag in a simultaneous translation system. They reorder the English text to more so mimic Japanese syntax, using techniques such as passivization. This could also ameliorate the ordering issues in our system. While this may reduce the necessity of verb prediction, it may be possible to interpolate between models trained on reordered corpora and non-reordered corpora by implementing actions for each.

- **Translation of Phrases** Currently, we send every observed word to the translation system at a commit action. Previous approaches (e.g., Fujita et al. (2013b)) have instead focused on segmenting at meaningful phrase units and translating them in turn. This approach would also ameliorate the issues with length and reordering. One possible approach is to let the system learn which strategy to use. It is possible to have both a **commit phrase** action and a **commit all** action. The former would append the most recent phrase translation to the extant output; the latter would use our current method of sending all observed input to the translation system. Supporting multiple translation hypotheses and incremental decoding (Sankaran et al., 2010) would improve both the efficiency and effectiveness of our system. Using data from human translators (Shimizu et al., 2014) could also add richer strategies for simultaneous translation: passive constructions, reordering, etc.

These and other shortcomings must be addressed before our system can be used in a real setting. We have, however, shown that the framework itself is viable and generalizable.

Chapter 6

Conclusion

In this dissertation, I have described the small steps we have taken toward prediction- and reinforcement learning-based simultaneous machine translation for languages with divergent word order: I have described results which strongly suggest that reinforcement learning with prediction is a viable approach to SOV-SVO simultaneous machine translation; I have described a metric, LBLEU, for incrementally quantifying translation quality; I have described a method for predicting final verbs in SOV languages and provided a human benchmark against which to compare; and I shown that these components can be combined into a translation system 2that learns to improve its simultaneous translations.

6.0.1 Future Directions

Improved Translation and Learning Models

A shortcoming of our approach is that SEARN requires a teaching policy. Finding an optimal policy becomes more and more computationally expensive with each added action. It is worth exploring other models that do not require this kind of teacher to learn. Since our work in Chapter 4, there has already arisen important work on using deep reinforcement learning for simultaneous machine translation (Cho and Esipova, 2016; Gu et al., 2017). This is appealing because it combines translation, decoding, decision-making, and prediction into a single architecture. Some work (Cho and Esipova, 2016) suggests that verb prediction is implicit in the deep neural model, but I believe that more research is necessary to determine whether this is true, given what we have learned about the difficulty and subtlety of verb prediction: we cannot count on a language model, even a good one, to predict verbs well. This is noted in subsequent work. (Gu et al., 2017)

Repairing and Evaluation

What should the system do when it makes a mistake? Currently, the system simply continues. This may be a particularly troublesome shortcoming given that we send all current observations to the decoder, rather than segmenting by phrase. In a true speech-to-speech system, there can be no deletions, but we must determine some way of making repairs, perhaps mimicking the behavior of simultaneous interpreters. The problem, however, is one of evaluation. Our current evaluation metrics are not designed to handle disfluencies. As noted in Section 2.5, BLEU is known to have serious problems with Japanese-English translation as it is. We need to learn how humans evaluate simultaneous translations, so that we can design metrics that take correlate with their judgments. For the Japanese case, we could, for example, incrementalize RIBES as we did with BLEU.

Expanded and More Robust Prediction

We have studied incremental verb prediction for German and Japanese using surface-level features represented as n-grams. We do not make use of any explicit parse information or external lexical databases. Given the apparent importance of arguments in human processing as shown in previous psycholinguistic research (Section 3.1), and given that case information was shown to be helpful for both humans and machines in both German and Japanese (Sections 3.2 and 3.3), it is reasonable to believe that the same may hold true for, for instance, argument information. There already exists work on Japanese-English translation with incremental dependency parsing (Ohno et al., 2005; Ryu et al., 2006), but no work on how it might aid verb prediction directly.

While we have focused on verb prediction, we have not dealt with the prediction of other late-occurring elements. In Japanese, for instance, negation, tense, aspect, and a host of other important information occurs as either a post-stem verbal inflection or immediately following the verb phrase. We focus on entire sentences, but it may be more useful to predict at the clause level, and perhaps easier. Relative clauses are one case; entire independent clauses—for example, separated by a conjunction—are another. Similarly, we have not investigated whether we can predict when the sentence (or the clause) will end. All of this could be useful for a system to learn to make its decisions.

蝸牛そろそろ登れ富士の山
katatsumuri sorosoro nobore fuji no yama
O snail,
Little by little,
Climb Mount Fuji.

(Kobayashi Issa, 1763-1868)

Bibliography

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In Proceedings of the International Conference of Machine Learning.

Abekawa, T. and Okumura, M. (2005). Corpus-based analysis of Japanese relative clause constructions. In International Conference on Natural Language Processing, pages 46–57. Springer.

Backhouse, A. E. (1984). Have all the adjectives gone? Lingua, 62(3):169–186.

Baker, M. (2003). Verbal adjectives as adjectives without phi-features. In Tokyo Conference on Psycholinguistics, pages 1–22. Keio University.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the Association for Computational Linguistics. Association for Computational Linguistics.

Bevilacqua, L. (2009). The position of the verb in Germanic languages and simultaneous interpretation.

Białecki, A., Muir, R., Ingersoll, G., and Imagination, L. (2012). Apache Lucene 4. In SIGIR, page 17.

Biemann, C., Heyer, G., Quasthoff, U., and Richter, M. (2007). The leipzig corpora collection-monolingual corpora of standard size. Proceedings of Corpus Linguistics.

Bock, K. and Levelt, W. (2002). Language production. Psycholinguistics: Critical concepts in psychology, 5:405.

Boyd-Graber, J., Satinoff, B., He, H., and Daumé III, H. (2012). Besting the quiz master: Crowdsourcing incremental classification games. In Conference of Empirical Methods in Natural Language Processing, pages 1290–1301. Association for Computational Linguistics.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In Conference of Empirical Methods in Natural Language Processing. Citeseer.

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.

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

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The salsa corpus: a German corpus resource for lexical semantics. In International Language Resources and Evaluation, pages 969–974.

Chesterman, A. (2016). Memes of translation: The spread of ideas in translation theory, volume 123. John Benjamins Publishing Company.

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

Cho, K. and Esipova, M. (2016). Can neural machine translation do simultaneous translation? arXiv preprint arXiv:1606.02012.

Chow, W.-Y. and Phillips, C. (2013). No semantic illusions in the "semantic p600" phenomenon: Erp evidence from mandarin chinese. Brain research, 1506:76–93.

Chow, W.-Y., Smith, C., Lau, E., and Phillips, C. (2015). A "bag-of-arguments" mechanism for initial verb predictions. Language, Cognition and Neuroscience, pages 1–20.

Christoffels, I. K. (2004). Cognitive studies in simultaneous interpreting. PhD thesis, University of Amsterdam.

Crammer, K., Kulesza, A., and Dredze, M. (2013). Adaptive regularization of weight vectors. Machine Learning, 91(2):155–187.

Daumé, H., Langford, J., and Marcu, D. (2009). Search-based structured prediction. Machine learning, 75(3):297–325.

Daumé III, H., Langford, J., and Marcu, D. (2006). Searn in practice. Unpublished (available at http://pub. hal3. name/).

Den, Y. and Inoue, M. (1997). Disambiguation with verb-predictability: Evidence from Japanese garden-path phenomena. In Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society, pages 179–184. Lawrence Erlbaum.

Deng, Y. and Byrne, W. (2008). Hmm word and phrase alignment for statistical machine translation. IEEE Transactions on Audio, Speech, and Language Processing, 16(3):494–507.

Dennett, D. C. (1995). Darwin's dangerous idea. The Sciences, 35(3):34–40.

Denoual, E. and Lepage, Y. (2005). Bleu in characters: towards automatic mt evaluation in languages without word delimiters. In International Joint Conference on Natural Language Processing, pages 81–86.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In International Conference on Human Language Technology Research, pages 138–145. Morgan Kaufmann Publishers Inc.

Dostert, L. E. (1955). The Georgetown-ibm experiment. Machine translation of languages, pages 124–135.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. Machine learning, 7(2-3):195–225.

Ferreira, F. (2000). Syntax in language production: An approach using tree-adjoining grammars. Aspects of language production, pages 291–330.

Flannery, D., Miyao, Y., Neubig, G., and Mori, S. (2012). A pointwise approach to training dependency parsers from partially annotated corpora. Information and Media Technologies, 7(4):1489–1513.

Frank, S. L., Otten, L. J., Galli, G., and Vigliocco, G. (2013). Word surprisal predicts n400 amplitude during reading. In Proceedings of the Association for Computational Linguistics.

Friederici, A. D. and Frisch, S. (2000). Verb argument structure processing: The role of verb-specific and argument-specific information. Journal of Memory and Language, 43(3):476–507.

Fügen, C. (2008). A system for simultaneous translation of lectures and speeches. PhD thesis, KIT-Bibliothek.

Fujita, T., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2013a). Simple, lexicalized choice of translation timing for simultaneous speech translation. INTERSPEECH.

Fujita, T., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2013b). Simple, lexicalized choice of translation timing for simultaneous speech translation. In Proceedings of Interspeech.

Furuse, O. and Iida, H. (1996). Incremental translation utilizing constituent boundary patterns. In Conference on Computational linguistics, pages 412–417. Association for Computational Linguistics.

Gaiba, F. (1998). The origins of simultaneous interpretation: The Nuremberg Trial. University of Ottawa Press.

Garvin, P. L. (1968). The Georgetown-IBM experiment of 1954: an evaluation in retrospect. Mouton.

Gile, D. (2009). Basic concepts and models for interpreter and translator training, volume 8. John Benjamins Publishing.

Goldman-Eisler, F. (1972). Segmentation of input in simultaneous translation. Journal of Psycholinguistic Research, 1(2):127–140.

Goto, I., Chow, K. P., Lu, B., Sumita, E., and Tsou, B. K. (2013). Overview of the patent machine translation task at the ntcir-10 workshop. In NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-10.

Grissom II, A., Orita, N., and Boyd-Graber, J. (2016). Incremental prediction of sentence-final verbs. In Conference on Computational Natural Language Learning.

Grissom II, A. C., He, H., Boyd-Graber, J., Morgan, J., and Daumé III, H. (2014). Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In Empirical Methods in Natural Language Processing.

Gu, J., Neubig, G., Cho, K., and Li, V. O. (2017). Learning to translate in real-time with neural machine translation.

Hale, J. (2001). A probabilistic earley parser as a psycholinguistic model. In Conference of the North American Chapter of the Association for Computational Linguistics.

Hale, J. (2003). The information conveyed by words in sentences. Journal of Psycholinguistic Research, 32(2):101–123.

Harris, B. (1988). Bi-text, a new concept in translation theory. Language Monthly, 54:8–10.

He, H., Boyd-Graber, J., Graber, J. B., and Daumé III, H. (2016). Interpretese vs. translationese: The uniqueness of human strategies in simultaneous interpretation. In Conference of the North American Chapter of the Association for Computational Linguistics, pages 971–976.

He, H., Grissom II, A., Boyd-Graber, J., and Daumé III, H. (2015). Syntax-based rewriting for simultaneous machine translation. In Conference of Empirical Methods in Natural Language Processing.

Hoeks, J. C., Stowe, L. A., and Doedens, G. (2004). Seeing words in context: the interaction of lexical and sentence level information during reading. Cognitive brain research, 19(1):59–73.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. (2010). Automatic evaluation of translation quality for distant language pairs. In Conference of Empirical Methods in Natural Language Processing, pages 944–952. Association for Computational Linguistics.

Iyyer, M., Boyd-Graber, J. L., Claudino, L. M. B., Socher, R., and Daumé III, H. (2014). A neural network for factoid question answering over paragraphs. In Conference of Empirical Methods in Natural Language Processing, pages 633–644.

Jakobson, R. (1959). On linguistic aspects of translation. On translation, 3:30–39.

Jörg, U. (1997). Bridging the gap: Verb anticipation in German-English simultaneous interpreting. Benjamins Translation Library, 20:217–228.

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. Cognitive Science, 20(2):137–194.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101(1):99–134.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, pages 237–285.

Kaji, N., Kawahara, D., Kurohash, S., and Sato, S. (2002). Verb paraphrase based on case frame alignment. In Proceedings of the Association for Computational Linguistics, pages 215–222. Association for Computational Linguistics.

Kamide, Y., Altmann, G., and Haywood, S. L. (2003). The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye movements. Journal of Memory and Language, 49(1):133–156.

Kawahara, D., Peterson, D., and Palmer, M. (2014). A step-wise usage-based method for inducing polysemy-aware verb classes. In Proceedings of the Association for Computational Linguistics, pages 1030–1040.

Kempen, G. and Huijbers, P. (1983). The lexicalization process in sentence production and naming: Indirect election of words. Cognition, 14(2):185–209.

Kiani, R. and Shadlen, M. N. (2009). Representation of confidence associated with a decision by neurons in the parietal cortex. Science, 324(5928):759–764.

Kneser, R. and Ney, H. (1995). Improved backing-off for n-gram language modeling. In Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. IEEE.

Knill, D. C. and Pouget, A. (2004). The bayesian brain: the role of uncertainty in neural coding and computation. TRENDS in Neurosciences, 27(12):712–719.

Koehn, P. (2000). German-English parallel corpus “de-news”.

Koehn, P. (2009). Statistical Machine Translation. Cambridge University Press.

Kolk, H. H., Chwilla, D. J., Van Herten, M., and Oor, P. J. (2003). Structure and limited capacity in verbal working memory: A study with event-related potentials. Brain and language, 85(1):1–36.

Konieczny, L. and Döring, P. (2003). Anticipation of clause-final heads: Evidence from eye-tracking and srns. In ICCS/ASCS.

Koso, A., Ojima, S., and Hagiwara, H. (2011). An event-related potential investigation of lexical pitch-accent processing in auditory Japanese. Brain research, 1385:217–228.

Kudo, T. (2005). Mecab: Yet another part-of-speech and morphological analyzer. http://mecab. sourceforge. net/.

Kurohashi, S. and Nagao, M. (1994). Kn parser: Japanese dependency/case structure analyzer. In Proceedings of the Workshop on Sharable Natural Language Resources.

Kutas, M., DeLong, K. A., and Smith, N. J. (2011). A look around at what lies ahead: prediction and predictability in language processing. Predictions in the brain: Using our past to generate a future, pages 190–207.

Lambert, S. (1991). Aptitude testing for simultaneous interpretation at the university of ottawa. Meta: Journal des traducteurs/Meta: Translators’ Journal, 36(4):586–594.

Langford, J., Li, L., and Strehl, A. (2007). Vowpal wabbit online learning project.

Langford, J. and Zadrozny, B. (2005). Relating reinforcement learning performance to classification performance. In Proceedings of the International Conference of Machine Learning.

Lau, E., Almeida, D., Hines, P. C., and Poeppel, D. (2009). A lexical basis for n400 context effects: Evidence from meg. Brain and language, 111(3):161–172.

Le Féal, D. (1990). ’some thoughts on the evaluation of simultaneous interpretation.’. D. and M. Bowen, eds. Interpreting–Yesterday, today and tomorrow, Binghamton, NY: SUNY, pages 154–60.

Levy, R. P. and Keller, F. (2013). Expectation and locality effects in German verb-final structures. Journal of memory and language, 68(2):199–222.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In International Language Resources and Evaluation.

Lo, C.-k. and Wu, D. (2011). Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In Proceedings of the Association for Computational Linguistics, pages 220–229. Association for Computational Linguistics.

Marslen-Wilson, W. (1973). Linguistic structure and speech shadowing at very short latencies. Nature, 244(5417speech perception of syntactic and semantic structures vs individual syllables and response latency, passage content memory, close vs more distant shadowers):522–523+.

Maruyama, T., Kashiok, T., and Tanaka, H. (2004). Development and evaluation of Japanese clause boundaries annotation program. Jornal of Natural Language Processing (in Japanese), 11(3):39–68.

Matsubara, S., Iwashima, K., Kawaguchi, N., Toyama, K., and Inagaki, Y. (2000a). Simultaneous Japanese-English interpretation based on early predition of English verb. In Symposium on Natural Language Processing.

Matsubara, S., Iwashimaz, K., Kawaguchizx, N., Toyama, K., and Inagaki, Y. (2000b). Simultaneous Japanese-English interpretation based on early prediction of English verb.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Conference on Neural Information Processing Systems, pages 3111–3119.

Miller, G. A. and Isard, S. (1963). Some perceptual consequences of linguistic rules. Journal of Verbal Learning and Verbal Behavior, 2(3):217 – 228.

Mima, H., Iida, H., and Furuse, O. (1998a). Simultaneous interpretation utilizing example-based incremental transfer. In International Conference on Computational Linguistics, pages 855–861. Association for Computational Linguistics.

Mima, H., Iida, H., and Furuse, O. (1998b). Simultaneous interpretation utilizing example-based incremental transfer. In Proceedings of the Association for Computational Linguistics.

Miyamoto, E. T. and Nakamura, M. (2013). Unmet expectations in the comprehension of relative clauses in Japanese. In CogSci.

Momma, S. (2016). Parsing, Generation and Grammar. PhD thesis, University of Maryland, College Park.

Momma, S., Slevc, L., and Phillips, C. (2015). The timing of verb planning in active and passive sentence production. In CUNY Conference on Human Sentence Processing.

Müller, M., Nguyen, T. S., Niehues, J., Cho, E., Krüger, B., Ha, T.-L., Kilgour, K., Sperber, M., Mediani, M., Stüker, S., et al. (2016). Lecture translator speech translation framework for simultaneous lecture translation. Conference of the North American Chapter of the Association for Computational Linguistics, page 82.

Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. Artificial and human intelligence, pages 351–354.

Namai, K. (2002). The word status of Japanese adjectives. Linguistic Inquiry, pages 340–349.

Neubig, G. (2011). The Kyoto free translation task. Available online at http://www.phontron.com/kftt.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In International Language Resources and Evaluation, volume 6, pages 2216–2219.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In Proceedings of the Association for Computational Linguistics, pages 440–447. Association for Computational Linguistics.

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

Och, F. J., Tillmann, C., Ney, H., et al. (1999). Improved alignment models for statistical machine translation. In Conference of Empirical Methods in Natural Language Processing, pages 20–28.

Oda, Y., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2014). Optimizing segmentation strategies for simultaneous speech translation. In Proceedings of the Association for Computational Linguistics.

Oda, Y., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2015). Syntax-based simultaneous translation through prediction of unseen syntactic constituents. Proceedings of the Association for Computational Linguistics.

Ohno, T., Matsubara, S., Kashioka, H., Kato, N., and Inagaki, Y. (2005). Incremental dependency parsing of Japanese spoken monologue based on clause boundaries. In European Conference on Speech Communication and Technology.

Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. Mathematics of operations research, 12(3):441–450.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002a). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the Association for Computational Linguistics, pages 311–318. Association for Computational Linguistics.

Peterson, D. W., Boyd-Graber, J., Palmer, M., and Kawhara, D. (2016). Leveraging verbnet to build corpus-specific verb clusters. c2016 The* SEM 2016 Organizing Committee. All papers c2016 their respective authors. This proceedings volume and all papers therein are licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons. org/licenses/by/4.0, page 102.

Pöchhacker, F. (2001). Quality assessment in conference and community interpreting. Meta: Journal des traducteurs, 46(2):410–425.

Popović, M., Stein, D., and Ney, H. (2006). Statistical machine translation of German compound words. In Advances in Natural Language Processing, pages 616–624. Springer.

Post, M., Cao, Y., and Kumar, G. (2015). Joshua 6: A phrase-based and hierarchical statistical machine translation system. The Prague Bulletin of Mathematical Linguistics, 104(1):5–16.

Pytlik, B. and Yarowsky, D. (2006). Machine translation for languages lacking bitext via multilingual gloss transduction. In Association for Machine Translation in the Americas.

Quasthoff, U., Richter, M., and Biemann, C. (2006). Corpus portal for search in monolingual corpora. In International Language Resources and Evaluation, pages 1799–1802.

Ramler, S. and Berry, P. (2009). Nuremberg and Beyond: The Memoirs of Siegfried Ramler from 20th Century Europe to Hawai'i. Booklines Hawaii Limited.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. Journal of machine learning research, 5(Jan):101–141.

Ryu, K., Matsubara, S., and Inagaki, Y. (2006). Simultaneous English-Japanese spoken language translation based on incremental dependency parsing and transfer. In Proceedings of the Association for Computational Linguistics.

Ryu, K., Matsubara, S., and Inagaki, Y. (2012). Alignment-based translation unit for simultaneous Japanese-English spoken dialogue translation. In Innovations in Intelligent Machines, pages 33–44. Springer.

Sakamoto, A., Watanabe, N., Kamatani, S., and Sumita, K. (2013). Development of a simultaneous interpretation system for face-to-face services and its evaluation experiment in real situation.

Sankaran, B., Grewal, A., and Sarkar, A. (2010). Incremental decoding for phrase-based statistical machine translation. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation.

Schank, R. C. (1983). Dynamic memory: A theory of reminding and learning in computers and people. cambridge university press.

Schriefers, H., Teruel, E., and Meinshausen, R. (1998). Producing simple sentences: Results from picture–word interference experiments. Journal of Memory and Language, 39(4):609–632.

Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon. PhD thesis, University of Pennsylvania.

Sennrich, R., Schneider, G., Volk, M., and Warin, M. (2009). A new hybrid dependency parser for German. German Society for Computational Linguistics and Language Technology, 115:124.

Shannon, C. E. (1948). A mathematical theory of communication. Bell Systems Technical Journal, 27:379–423, 623–656.

Shimizu, H., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2014). Collection of a simultaneous translation corpus for comparative analysis. In International Language Resources and Evaluation.

Skadiņš, R., Tiedemann, J., Rozis, R., and Deksne, D. (2014). Billions of parallel words for free: Building and using the eu bookshop corpus. In International Language Resources and Evaluation, Reykjavik, Iceland. European Language Resources Association (ELRA).

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In In Proceedings of Association for Machine Translation in the Americas.

Solomonoff, R. J. (1957). An inductive inference machine. In IRE Convention Record, Section on Information Theory, volume 2, pages 56–62.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. arXiv preprint cs/0609058.

Sumita, E., Iida, H., and Kohyama, H. (1990). Translating with examples: a new approach to machine translation. In Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, number 3, pages 203–212.

Syed, U., Bowling, M., and Schapire, R. E. (2008). Apprenticeship learning using linear programming. In Proceedings of the International Conference of Machine Learning.

Syed, U. and Schapire, R. E. (2010). A reduction from apprenticeship learning to classification. In Conference on Neural Information Processing Systems, pages 2253–2261.

Takagi, A., Matsubara, S., Kawaguchi, N., and Inagaki, Y. (2002). A corpus-based analysis of simultaneous interpretation. In International Joint Conference on Natural Language Processing, pages 167–174.

Tanaka, Y. (2001). Compilation of a multilingual parallel corpus. PACLING, pages 265–268.

Tewari, A. and Bartlett, P. L. (2007). On the consistency of multiclass classification methods. Journal of Machine Learning Research, 8(May):1007–1025.

Thrun, S. and Littman, M. L. (2000). A review of reinforcement learning. AI Magazine, 21(1):103–105.

Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, Recent Advances in Natural Language Processing, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, International Language Resources and Evaluation, Istanbul, Turkey. European Language Resources Association (ELRA).

Tillmann, C., Vogel, S., Ney, H., and Zubiaga, A. (1997). A dp-based search using monotone alignments in statistical translation. In Proceedings of the Association for Computational Linguistics.

Tohyama, H. and Matsubara, S. (2006). Collection of simultaneous interpreting patterns by using bilingual spoken monologue corpus. In International Language Resources and Evaluation.

Toutanova, K., Ilhan, H. T., and Manning, C. D. (2002). Extensions to hmm-based statistical word alignment models. In Conference of Empirical Methods in Natural Language Processing, pages 87–94. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Conference of the North American Chapter of the Association for Computational Linguistics, pages 173–180.

Uchimoto, K., Sekine, S., and Isahara, H. (1999). Japanese dependency structure analysis based on maximum entropy models. In Proceedings of the Association for Computational Linguistics, pages 196–203. Association for Computational Linguistics.

Van Berkum, J. J., Brown, C. M., Zwitserlood, P., Kooijman, V., and Hagoort, P. (2005). Anticipating upcoming words in discourse: evidence from erps and reading times. Journal of Experimental Psychology: Learning, Memory, and Cognition, 31(3):443.

Van Herten, M., Chwilla, D. J., and Kolk, H. H. (2006). When heuristics clash with parsing routines: Erp evidence for conflict monitoring in sentence perception. Journal of cognitive neuroscience, 18(7):1181–1197.

Venuti, L. (2008). The translator's invisibility: A history of translation. Routledge.

Vogel, S., Ney, H., and Tillmann, C. (1996a). Hmm-based word alignment in statistical translation. In Conference on Computational linguistics, pages 836–841. Association for Computational Linguistics.

Vogel, S., Ney, H., and Tillmann, C. (1996b). HMM-based word alignment in statistical translation. In Proceedings of International Conference on Computational Linguistics.

Wahlster, W. (2000). Verbmobil: foundations of speech-to-speech translation. Springer.

Wang, H., Ding, C. H., and Huang, H. (2010). Multi-label classification: Inconsistency and class balanced k-nearest neighbor. In AAAI.

Watanabe, Y., Asahara, M., and Matsumoto, Y. (2010). A structured model for joint learning of argument roles and predicate senses. In Proceedings of the Association for Computational Linguistics, pages 98–102. Association for Computational Linguistics.

Weaver, W. (1955). Translation. Machine translation of languages, 14:15–23.

Wetzer, H. (1996). The typology of adjectival predication, volume 17. Walter de Gruyter.

Yamashita, H. (1994). Processing of Japanese and korean. Unpublished doctoral dissertation, Columbus, OH: Ohio State University.

Yamashita, H. (1997). The effects of word-order and case marking information on the processing of Japanese. Journal of Psycholinguistic Research, 26(2):163–188.

Yamashita, H. (2000). Structural computation and the role of morphological markings in the processing of Japanese. Language and speech, 43(4):429–455.

Yang, M. and Kirchhoff, K. (2006). Phrase-based backoff models for machine translation of highly inflected languages. In European Chapter of the Association for Computational Linguistics, pages 3–7.

Yoshikawa, M., Matsubara, S., Ryu, K., and Ding, Z. (2005). Interpreting unit segmentation of conversational speech in simultaneous interpretation corpus. In O-COCOSDA 2005, pages 148–152. The oriental chapter of COCOSDA (The International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques).