

Copyright
by
Alexis Mary Palmer
2009

The Dissertation Committee for Alexis Mary Palmer
certifies that this is the approved version of the following dissertation:

**Semi-Automated Annotation and Active Learning
for Language Documentation**

Committee:

Jason Baldrige, Supervisor

Katrin Erk, Supervisor

Nora England

Raymond Mooney

Anthony Woodbury

**Semi-Automated Annotation and Active Learning
for Language Documentation**

by

Alexis Mary Palmer, B.A.;M.A.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2009

for CSS
You inspire me still.

Oh! The places you'll go...

Dr. Seuss

Acknowledgments

Graduate students often commiserate over how mentally and emotionally taxing the dissertation process can be. When I first heard senior students voicing such sentiments, I remember wondering what the fuss was all about. Sure, it's a lot of work, I thought, but must it be so stressful? I have since come to see crossing the stormy seas of researching and writing a dissertation as a marathon of intellectual and personal growth. I find myself at last standing on what was once the far shore, a stronger, smarter, more engaged individual. And I have many people to thank for their support along the way.¹

I am enormously fortunate to have had Katrin Erk and Jason Baldrige as co-supervisors. Both Jason and Katrin have consistently provided guidance, inspiration, and motivation, all with great patience, understanding, and good humor. Not to mention that they're both really cool people who I'm happy to have as friends. Carlota Smith, to whom this work is dedicated, will forever stand as a mentor and role model for me, personally as well as academically. Carlota showed me what it looks like to work at something you truly love, taking the difficulties in stride and focusing instead on the joys of intellectual discovery, both as a teacher and as a researcher. Special thanks to Jonas Kuhn for supervising my Master's thesis, and for all that I learned along the way.

¹I am certain to have inadvertently missed some very important people; if you are one of those people, please accept my apology, and the next coffee is on me!

And finally, thanks to Caroline Sporleder for support in the final stages of thesis writing and defense preparation.

This research would not have been possible without the tremendous annotation efforts of Eric Campbell and Telma Can, nor without collaboration with Taesun Moon, in particular development of the annotation tool. B'alam Mateo-Toledo has long been a sounding board for my ideas about data management and automated processing for documentation. He was also instrumental, along with Nora England and OKMA, in providing access to the data used in this research. Andrew Garrett, Taesun Moon, Elias Ponvert, Mike Rich, and Jody Saks helped with feedback and proofreading of the thesis itself. Nick Gaylord, Andrew Harrison, and Emmy Destruel provided *crucial* logistical assistance. And Eric McCready deserves special thanks for convincing me that yes, computational linguistics and indigenous languages *do* have something to do with each other.

The UT Department of Linguistics is special in that it gives students the opportunity to explore many different aspects of linguistics and linguistic theory, as well as to learn from an outstanding faculty. Of particular note in my education are David Beaver, John Beavers, Rajesh Bhatt, Megan Crowhurst, Nora England, Ray Mooney (Computer Science), Bernhard Schwarz, Steve Wechsler, Tony Woodbury, and Roberto Zavala. I am grateful also for the many students willing to share their insights regarding language documentation: especially Lynda Boudreault, Emiliana Cruz, Hilaria Cruz, Susan Kung, Liberty Lidz, B'alam Mateo-Toledo, and Christina Willis. Outside the de-

partment, I have learned a great deal from discussions with (among others) Emily Bender, Steven Bird, Jeff Good, Heidi Johnson, Dan Jurafsky, Katrin Tomanek, Will Lewis, Mark Liberman, Martha Palmer, Burr Settles, and Nick Thieberger.

My wonderful friends and family have provided support of all kinds. In particular I must mention my Croatians—Thomas, Marija, Ana, Žarko, and Andrija—and the other members of my Austin family: Michael, JR, Kristi, John, Gabrelle, Kristen, Ron, and Leslie. Cale, you were there for some of the stormiest seas... thank you for your love and friendship. Mike, thank you for believing in me. My linguist friends: Aaron, Alex, Anne, B'alam, Brian, Cynthia, Doug, Eric C., Eric M., Fred, Jess, Lynda, Pascal, Sadaf, Susan, Telma, and so many others. Are all linguists amazing people, or did we just get really lucky? Thanks to my fellow members of the Austin Handbell Ensemble and the Austin Symphonic Band for sharing the wonderful restorative power of making music together. Finally, many thanks to my new friends in Saarbrücken who have celebrated with me as this journey at last comes to an end.

And Mom, Dad, Gary, Zach, Sara, all the Palmers and Goldens and Hofbauers and Thomas and Saks... thank you for your never-ending love and support.

Semi-Automated Annotation and Active Learning for Language Documentation

Publication No. _____

Alexis Mary Palmer, Ph.D.
The University of Texas at Austin, 2009

Supervisors: Jason Baldrige
Katrin Erk

By the end of this century, half of the approximately 6000 extant languages will cease to be transmitted from one generation to the next. The field of **language documentation** seeks to make a record of endangered languages before they reach the point of the extinction, while they are still in use. The work of documenting and describing a language is difficult and extremely time-consuming, and resources are extremely limited. Developing efficient methods for making lasting records of languages may increase the amount of documentation achieved within budget restrictions.

This thesis approaches the problem from the perspective of **computational linguistics**, asking whether and how automated language processing can reduce human annotation effort when very little labeled data is available for model training. The task addressed is morpheme labeling for the Mayan

language Uspanteko, and we test the effectiveness of two complementary types of machine support: (a) learner-guided selection of examples for annotation (active learning); and (b) annotator access to the predictions of the learned model (semi-automated annotation).

Active learning (AL) has been shown to increase efficacy of annotation effort for many different tasks. Most of the reported results, however, are from studies which simulate annotation, often assuming a single oracle that always provides accurate labels. In our studies, crucially, annotation is *not* simulated but rather performed by human annotators. We measure and record the time spent on each annotation, which in turn allows us to evaluate the effectiveness of machine support in terms of *actual* annotation effort.

We report three main findings with respect to active learning. First, in order for efficiency gains reported from active learning to be meaningful for realistic annotation scenarios, the type of cost measurement used to gauge those gains must faithfully reflect actual annotation cost. Second, the relative effectiveness of different selection strategies in AL seems to depend in part on the characteristics of the annotator, so it is important to model the individual oracle or annotator when choosing a selection strategy. And third, the cost of labeling a given instance from a sample is not a static value but rather depends on the context in which it is labeled.

We report two main findings with respect to semi-automated annotation. First, machine label suggestions have the potential to increase annotator efficacy, but the degree of their impact varies by annotator, with annotator

expertise a likely contributing factor. At the same time, we find that implementation and interface must be handled very carefully if we are to accurately measure gains from semi-automated annotation.

Together these findings suggest that simulated annotation studies fail to model crucial human factors inherent to applying machine learning strategies in real annotation settings.

Table of Contents

Acknowledgments	v
Abstract	viii
List of Tables	xiv
List of Figures	xv
Chapter 1. Introduction	1
1.1 Research questions	2
1.2 Context of annotation	4
1.3 Machine-assisted annotation for IGT	7
1.3.1 Unlabeled data	7
1.3.2 Semi-automated annotation	8
1.3.3 Active learning	9
1.4 Active learning with live annotation	10
1.4.1 Measuring annotation cost	10
1.4.2 Modeling the annotator	11
1.5 Contributions	13
1.6 Structure of the dissertation	14
Chapter 2. Documenting and describing endangered languages	15
2.1 Language endangerment	15
2.1.1 What does it mean to call a language ‘endangered’?	17
2.1.2 Documentation, description, and maintenance	19
2.2 Language documentation process	22
2.3 Annotation as ongoing analysis	25
2.4 Computational linguistics and language documentation	27

Chapter 3. Interlinear glossed text	29
3.1 Definition and general description	29
3.1.1 IGT for language documentation	30
3.1.2 Variability in IGT	32
3.1.3 Machine-readable IGT	34
3.2 The IGT-XML format	35
3.2.1 Basics of IGT-XML	36
3.2.2 Details of the format	38
3.3 Related work	40
3.3.1 XML formats for IGT	41
3.3.2 Tools for IGT	42
3.3.3 Varied practices for IGT production	44
3.3.4 Automated production of IGT	45
Chapter 4. Data and data preparation	48
4.1 The corpus: Uspanteko texts	48
4.2 Data clean-up and conversion	50
4.2.1 Cooperative data clean-up	52
4.2.2 Error types and correction methods	54
4.3 Target representation	58
Chapter 5. Active learning with simulated annotation	61
5.1 Active learning	62
5.2 Organization of corpus for experimentation	66
5.3 Model and methods	70
5.4 Simulation experiments	72
5.4.1 Parameters and evaluation	72
5.4.2 Results	74

Chapter 6. Active learning with live annotation	78
6.1 Annotators	79
6.1.1 Annotator expertise	80
6.1.2 Annotator training and learning curve	81
6.2 Annotation infrastructure	82
6.2.1 Label suggestions	84
6.2.2 Annotation tool	85
6.3 Annotation experiments	88
6.3.1 Experimental conditions	88
6.3.2 Results	88
6.3.3 Discussion	94
Chapter 7. Complexities of active learning with live annotation	99
7.1 Annotator accuracy and consistency	99
7.2 Annotator agreement, with error analysis	102
7.2.1 The ESP error	103
7.3 Reflections on the annotation experience	105
7.3.1 Annotation tool	106
7.3.2 Labeling-retraining cycle	107
7.3.3 Iterative model development	108
7.3.4 Epiphany effect	110
7.3.5 Handling changes in analysis	111
Chapter 8. Conclusion	113
References	121
Vita	132

List of Tables

2.1	Evaluating language endangerment (UNESCO, 2003b)	18
3.1	Decomposition of GLOSS and POS tiers of IGT.	46
4.1	Hyphenation possibilities for a three morpheme word form. . .	56
4.2	Repeat: Decomposition of GLOSS and POS tiers of IGT. . . .	59
4.3	Most common labels and their frequencies.	60
5.1	Corpus divisions.	67
5.2	Genre balance in corpus divisions	68
5.3	Corpora: number of words and sentences, number of possible tags, and average sentence length.	69
5.4	Training on all data: unigram probability and model performance	71
5.5	OPER values for Uspanteko simulations, comparing clause and morpheme cost. $\frac{A}{B}$ indicates we compute OPER(A,B). . . .	75
5.6	OPER values for morpheme cost for simulations. $\frac{A}{B}$ indicates we compute OPER(A,B).	76
6.1	OPER for language expert over language novice.	92
6.2	OPER comparisons using time cost measurement, language expert.	93
6.3	OPER comparisons using time cost measurement, language novice. 93	
7.1	Overall accuracy of annotators' labels, measured against OKMA annotations.	100
7.2	Intra-annotator consistency, language expert	102
7.3	Intra-annotator consistency, language novice	102
7.4	IAA: LgExp v. LgNov , percentage of morphemes in agreement, (number of duplicate clauses)	103

List of Figures

3.1	IGT-XML: Uspanteko clause.	39
3.2	Toolbox output: Uspanteko clause	42
5.1	Learning curves for Uspanteko simulations; (a) clause cost and (b) morpheme cost.	75
6.1	Average annotation time (in seconds per morpheme) over annotation rounds, averaged over all six conditions for each annotator.	83
6.2	The OpenNLP IGT Editor interface.	86
6.3	A sample of the learning curves with (a) morpheme cost and (b) time cost.	90
6.4	Sample measurements and fitted nonlinear regression curves for (a) the LgExp and (b) the LgNov . Note that the scale is consistent for comparability.	91
6.5	Accuracy on previously-unseen morphemes for both annotators, seq vs. unc	96

Chapter 1

Introduction

Estimates of the number of languages spoken today vary, but most fall in the neighborhood of 6000-6500 signed and spoken languages. The majority of these have no written form. When a language has never been written down, the main resource for keeping a record of that language are its speakers. The record is maintained as new generations learn the language and use it as their primary means of communication, and the language dies along with its last speakers.¹

Languages are dying at an alarming rate, and by the end of this century, half of the approximately 6000 languages will cease to be transmitted from one generation to the next (Crystal, 2000). A language so endangered may continue to be used in limited domains, or by a small subset of people, but with no new speakers, even limited use is unlikely to last more than one generation.

Global awareness of language endangerment has increased efforts to document these languages while there are still speakers. The work of documenting and describing a language is difficult and time-consuming, and al-

¹This is just one possible way of conceptualizing language endangerment. See Chapter 2 for further discussion.

though language endangerment is widely acknowledged as an important social concern, only very limited resources are available in support of language documentation work. Developing efficient methods for making a lasting record of a language may increase the amount of documentation that can be accomplished within budget restrictions and perhaps also increase the number of languages documented before they are lost.

This thesis explores the potential of automated language processing to reduce the time cost of documentation. The focus is on collected texts that have been transcribed and then annotated with detailed linguistic information, specifically targeting the stage of linguistic analysis and annotation. We approach the problem from the perspective of computational linguistics.

1.1 Research questions

Much recent research in computational linguistics and natural language processing aims to minimize the amount of supervision given to machine learners. One line of work focuses on using only unlabeled data, and another aims instead to learn good models with less manually-labeled data. This thesis takes the second path, using two complementary strategies: **semi-automatic annotation** and **active learning**. Both strategies aim specifically to reduce annotation cost, giving them a shared concern with documentary linguistics.

Semi-automatic annotation. Most semi-automated annotation uses supervised models trained on labeled data, with more training data producing

more accurate models. The language documentation context poses a challenge for *any* supervised learning methods because of the inherent data scarcity. We ask whether and how semi-automated methods can speed up linguistic annotation for language documentation, proposing the following thesis:

Thesis one. *Supervised machine learning can be effectively used for semi-automated annotation even when very little data is available for model development.*

Active learning. Active learning (AL) is an iterative annotation process in which the learner guides selection of examples to be labeled as training material for the next round of annotation. However, most active learning research simulates the role of the annotator, leaving open the question of whether results from simulated studies hold with real (i.e. non-simulated) annotators. This question informs the second thesis:

Thesis two. *Simulated active learning in its current state fails to model crucial human factors inherent to using active learning in real annotation settings.*

Approach. The motivating research questions are investigated by evaluating the effectiveness of these two types of machine support, both separately and together. We also vary the annotator, thus evaluating the effectiveness of both active learning and semi-automated annotation with different levels of annotator expertise. Our findings with respect to these claims are discussed

in Chapter 6. First, though, a more detailed description of the annotation context is warranted.

1.2 Context of annotation

The task modeled in the current annotation scenario is the production of interlinear glossed text (IGT, see also Chapter 3). Detailed linguistic analysis and annotation of texts is one of the most time-consuming stages of the language documentation process, and IGT is one widely-used format for presenting and representing such analysis. IGT production is a complex process with many interrelated components; of these, we target morpheme glossing.

Interlinear glossed text. In its prototypical form, IGT consists of four aligned levels of annotation displayed one below the other. (1) illustrates with a short Uspanteko clause.² The MORPH line displays each word from the TEXT line segmented into its component morphemes. Immediately below, the GLOSS line shows glosses for each morpheme. The bottom line presents one or more translations of the text being annotated.

- (1) TEXT: Xelch li
- MORPH: x- el -ch li
- GLOSS: COM- salir -DIR DEM
- TRANS: **Spanish:** ‘Salio entonces.’ **English:** ‘Then he left.’

²KEY: COM=completive aspect, DEM=demonstrative, DIR=directional

IGT is a valuable resource for further study of an individual language, for cross-linguistic studies, for theoretical work, and perhaps for development of automated language processing tools. Many language documentation projects treat languages from families about which our general linguistic knowledge is minimal, and the recordings and documents which have been collected by language documentation projects are a largely untapped wealth of linguistic and typological data.

Another aspect of our work on IGT is the development of an XML format (Section 3.2) specifically for IGT. With this format, IGT can be represented and stored in a way that is easily machine readable and usable.

Uspanteko corpus. The primary dataset used in these experiments is a corpus of texts in the Mayan language Uspanteko (Can et al., 2007a). Uspanteko is a member of the K'ichee' branch of the Mayan language family and is spoken by approximately 1320 people in central Guatemala (Richards, 2003). The texts were collected, transcribed, translated into Spanish, and annotated as part of an OKMA³ language documentation project.

Data clean-up. Machine analysis generally requires data that is internally consistent with respect to annotation and alignment. To prepare the Uspanteko data for machine analysis, first addressing clean-up and reformatting of the data. Many of the errors found in the original corpus result from accidental

³<http://www.okma.org>

inconsistencies in human annotation, a fact that emphasizes the importance of enforcing consistency at the point of the original annotations. By this we mean *not* that the annotators should have been more careful, but rather that the annotation infrastructure should make it easy for annotators to produce self-consistent annotations. When it comes to finding and resolving such errors, we argue for a cooperative approach, with a computational linguist and a language expert working in tandem.

Full-text annotation. From one perspective, the ideal end result of language documentation is manual labeling of the entire collected corpus; that result may be desirable, but it is almost never feasible, given resource limitations. At the same time, a text with only partial annotation is of limited use. Keeping in mind the goal of full-text annotation, we take a two stage approach.

First, available resources are put toward manual annotation in order to produce training material for a learner, improving the efficacy of the resources via semi-automated annotation and active learning. The remainder (unlabeled) portion of the corpus is then labeled automatically. The natural next step is a second round of manual annotation correcting the output of the machine labeler, however that is outside the scope of this dissertation.

1.3 Machine-assisted annotation for IGT

The annotation scenario described above models a point in the documentation process at which unlabeled data (i.e. texts that have not been glossed) is available in the form of text transcriptions and translations. The goal is to produce labeled data (i.e. glossed texts) and, further, to do so more efficiently than with standard manual annotation.

1.3.1 Unlabeled data

Given the predominance of unlabeled data in the language documentation context, we might consider using unsupervised methods, which have achieved promising results in a number of natural language processing tasks. However, most work in unsupervised analysis tackles tasks that are already well-understood.

Working with a previously-unstudied language introduces a number of uncertainties. It is already known that unsupervised systems can perform quite differently when applied to different domains. When the new ‘domain’ is a new language, the system could be confronted with radically different ways of organizing information, and as yet there is no way to predict whether familiar models will behave as expected.

There may be very little known about the language or how its properties interact with the task at hand. Even if the syntax and morphotactics of the language are well-understood, these things do not necessarily predict how the language will behave under computational analysis. A simple example:

some popular unsupervised approaches to morphology induction and analysis work well for languages with straightforward concatenative morphology but are unable to handle phenomena like infixation or circumfixion (e.g. Morfessor (Creutz and Lagus (2007)) and Linguistica (Goldsmith (2001))). The templatic morphology of languages like Hebrew and Arabic is also challenging for unsupervised approaches (Snyder and Barzilay (2008); Poon et al. (2009)).

Given the many uncertainties surrounding use of unsupervised methods in unfamiliar languages, we opt to focus our efforts on supporting manual production of labeled data rather than trying to learn from only unlabeled data using unsupervised methods.

1.3.2 Semi-automated annotation

Thesis one. *Supervised machine learning can be effectively used for semi-automated annotation even when very little data is available for model development.*

In general, semi-automated annotation (Brants and Plaehn, 2000) uses automatic processing, and often machine learning, to support manual annotation by making some annotation decisions ahead of time. The automatic component may predict labels for individual instances, or it might reduce the number of options provided to the annotator. We examine the effectiveness of displaying predicted morpheme labels to the annotator, who then has the option to change any of the predicted labels.

Finding one. In theory, semi-automated annotation reduces effort by making many labeling decisions ahead of time. In practice, the effectiveness of this strategy appears to vary by annotator, and annotator expertise seems to be a contributing factor.

Finding two. When providing machine support to manual annotation, implementation of the annotation interface must be handled carefully. For example, time spent navigating an inefficient interface increases the time cost of annotating an individual instance, thereby reducing the gains seen from machine support.

1.3.3 Active learning

Standard annotation practice in language documentation is to start at the beginning of a text and work through it in order. For the purposes of developing training material for a machine learner, though, it has been shown that annotating instances in the order in which they appear in the corpus is less effective than randomly picking instances to be labeled (Baldrige and Osborne, 2008). Effectiveness here is defined as annotating all instances as accurately as possible given a finite budget.⁴ In many cases, allowing the learner to select instances for annotation is the most effective strategy. This is the key insight behind active learning.

⁴We assume the budget is not sufficient to support complete annotation of the corpus.

1.4 Active learning with live annotation

Thesis two. *Simulated active learning in its current state fails to model crucial human factors inherent to using AL in real annotation settings.*

The result that learner-guided selection reduces annotation effort was established in the context of simulated annotation. Before active learning can be recommended as a cost-savings strategy for projects involving human annotators, though, more must be understood about how learner-guided selection and human annotation interact.

1.4.1 Measuring annotation cost

In studies with simulated annotation, selected instances are labeled by retrieving gold-standard labels from the corpus. Annotation effort is usually measured with respect to the number of instances labeled, but this measure does not necessarily correspond to the effort required for a human annotator to label the instance.

Finding three. To accurately gauge the effectiveness of AL requires a cost measurement strategy that faithfully reflects actual annotation cost. In this thesis, time is used to represent annotation cost. Time is an important consideration for language documentation, and it is also an appropriate measure for many other annotation tasks.

In the live annotation experiments, annotation time is measured and recorded for every instance labeled. The results show clearly that unit cost measurements overstate cost savings from learner-guided selection. Time cost measurement does not, however, change the *relative* effectiveness of the different strategies.

Finding four. The cost to annotate a given instance is not a static value. First, the cost varies by annotator. Second, the cost varies, even for the *same annotator*, depending on the number and type of instances the annotator has previously labeled. In other words, annotation cost is meaningful only in context and is not invariant to permutations in the ordering of examples presented to the annotator for labeling—a perhaps obvious but often overlooked fact.

1.4.2 Modeling the annotator

Another mismatch between simulated and live annotation is that the former assumes a perfect annotator or oracle (Donmez and Carbonell, 2008). Human annotators in fact are not perfect, nor are they interchangeable.

The live annotation studies employ both an annotator with expertise in the language and one without. The latter represents one resource that might be available to a language documentation project: a linguist with time to give to the project but without prior experience in the language.

We find that the relative effectiveness of different sample selection strategies is different for a linguist annotator with expertise in the language

than it is for a novice linguist annotator. Results from the annotations of the language expert follow the patterns seen in simulation studies, with learner-guided selection getting the best results. For the language novice, though, random selection is just as effective as learner-guided selection. The latter method selects clauses which are very difficult for the novice to annotate, increasing the number of incorrect labels supplied by the novice. The models learned from this data are then less accurate.⁵

Finding five. The gains from using learner-guided sample selection instead of other selection methods depend on characteristics of the individual annotator. In particular, the annotator’s level of experience and expertise with the annotation task influences the relative effectiveness of different strategies. Thus it is important to model the individual oracle or annotator when choosing a selection strategy.

Having examined results from only two annotators, one language expert and one language novice, we are unable to draw conclusions about the best selection or suggestion strategies for a given type of annotator. What the results do show is that annotator expertise has a strong effect on the *relative* effectiveness of different strategies.

The results from applying active learning in live annotation are not uniform. For effective practical deployment of active learning, it is important to model the annotators and the annotation context.

⁵This result is discussed in more detail in Section 6.3.3.

1.5 Contributions

This dissertation contributes to active learning research as well as to work on semi-automated annotation. The key contributions made by this thesis are the following:

- In-depth studies of the efficacy of active learning and semi-automated annotation using actual human annotators. This is in contrast to most work in active learning, which tends to simulate annotation.
- Integrating machine predictions and learner-guided selection into the live annotation process is difficult in some unexpected ways. We provide a discussion of implementational concerns that turned out to influence the study results.
- An annotation tool for managing the interactions between the human annotator, the machine learner, and the process of selecting examples for annotation using different selection methods.
- A brief case study of corpus clean-up and transformation. In this process, a language documentation corpus was made more consistent as well as being prepared for machine analysis. The clean-up process benefited from combining the skills of a language expert with those of a computational linguist.
- A general, flexible XML format for storing/representing interlinear glossed text (IGT), an important linguistic data structure.

A less tangible impact of this dissertation is that it represents work that is grounded in and relevant for both computational linguistics and language documentation. At the moment, this intersection is sparsely populated (Bird, 2009), but the population is growing. Our results are cautiously promising: computational linguistics *can* help to speed up some aspects of language documentation, but we have to be careful and smart about how we do it.

1.6 Structure of the dissertation

The dissertation is roughly divided into two parts. Following the introduction, the next three chapters provide background and preliminaries to the studies presented in the second part. The chapters discuss language endangerment and documentation (Chapter 2), interlinear glossed text (Chapter 3), and issues concerning data and data clean-up (Chapter 4).

The next three chapters present our experiments in semi-automated annotation and active learning. Chapter 5 provides background on active learning and presents simulated annotation studies. Live annotation studies are discussed in Chapter 6, followed by a chapter on the complexities of integrating machine learning and manual annotation (Chapter 7). Finally Chapter 8 summarizes and suggests some possible directions for continuing work.

Chapter 2

Documenting and describing endangered languages

In this chapter we present some key issues in the documentation and description of the endangered languages of the world, including a discussion of the problem of language endangerment and some approaches to addressing the problem. After introducing the field, we focus on aspects of it that are particularly relevant for the design and interpretation of the experiments described in this thesis.¹

2.1 Language endangerment

Conservative estimates say that by the end of this century, half of the approximately 6000 extant spoken and signed languages will cease to be transmitted effectively from one generation of speakers to the next [Crystal,2000]. Increasingly over the last two decades, language loss has come to be seen as a globally-relevant problem. For example, the United Nations General Assembly declared 2008 to be the International Year of Languages. The declaration

¹This chapter includes material previously published in Palmer and Erk (2007) and contained in Palmer et al. (submitted).

makes a call to worldwide ‘stakeholders to develop, support and intensify activities aimed at fostering respect for and the promotion and protection of all languages, in particular endangered languages, linguistic diversity and multilingualism.’²

This declaration was largely in response to activities of the United Nations Educational, Scientific and Cultural Organization (UNESCO) and in particular its Division of Cultural Objects and Intangible Heritage. In 2003 this division released a document titled ‘Convention for the Safeguarding of the Intangible Cultural Heritage’ (UNESCO, 2003a) which lists as part of a group’s intangible cultural heritage ‘oral traditions and expressions, including language as a vehicle of the intangible cultural heritage’ (UNESCO, 2003a).³

In addition to being part of the cultural heritage of the community that speaks it, each individual language is a complete linguistic system with the potential to expand our understanding of the diversity of the world’s languages as well as the space of possible languages. The linguistics community has for nearly two decades been paying close attention to the fragile status of the majority of the world’s languages; Woodbury (2003) points to a 1991 LSA address (later published as Hale et al. (1992)) as a key galvanizing event. In addition, some of the earliest work in linguistics established a tradition of documenting and describing previously-unstudied languages. Thus the global

²<http://www.un.org/events/iyl/>

³UNESCO’s resources related to endangered languages can be found at <http://www.unesco.org/culture/ich/index.php?pg=00136>.

attention garnered by the UN declaration and similar activities has spread awareness of a long-standing concern.

2.1.1 What does it mean to call a language ‘endangered’?

There is no single accepted set of criteria for determining whether a language is ‘living’, ‘endangered’, ‘sleeping’, or ‘moribund’. It is clear, though, that simply counting the number of speakers of the language is not an accurate method for measuring whether a language is endangered or not. For example, a language spoken by 2000 people in an isolated village may be far more stable than one spoken by 20,000 but with strong social pressure to adopt a majority language.⁴

One way of evaluating endangerment status for languages is laid out in a report from an ad hoc UNESCO expert group on endangered languages (UNESCO, 2003b). The report defines six levels of endangerment: *safe*, *unsafe* (or *vulnerable*), *definitely endangered*, *severely endangered*, *critically endangered*, and *extinct*. The report discusses nine factors relevant for determining an individual language’s degree of endangerment, shown in Table 2.1.

Himmelman (2008) makes a strong argument against this factorized approach to evaluating the status of a language. The factors shown in Table 2.1 attempt to quantify and assess properties which are neither independent of one another nor disconnected from the context in which the language

⁴A similar argument appears in Crystal (2000).

1	Intergenerational Language Transmission
2	Absolute Number of Speakers
3	Proportion of Speakers within the Total Population
4	Trends in Existing Language Domains
5	Response to New Domains and Media
6	Materials for Language Education and Literacy
7	Governmental and Institutional Language Attitudes and Policies Including Official Status and Use
8	Community Members' Attitudes toward their Own Language
9	Amount and Quality of Documentation

Table 2.1: Evaluating language endangerment (UNESCO, 2003b)

is used. Rather, as argued by Himmelmann (1998); Woodbury (2003); Dobrin et al. (2009) and others, each language needs to be seen as part of a complex linguistic ecology, and work on documenting languages must be tailored to the individual situation. Nevertheless, the UNESCO report provides an interesting introduction to some less-widely-considered aspects of language endangerment and preservation.

In this work we adopt very general notions of language endangerment, taken from introductory material in the report (UNESCO, 2003b)[2]:

A language is *endangered* when it is on a path toward extinction. Without adequate documentation, a language that is extinct can never be revived.

A language is in danger when its speakers cease to use it, use it in an increasingly reduced number of communicative domains, and cease to pass it on from one generation to the next. That is, there are no new speakers, adults or children.

The work presented in this thesis was designed with special consideration for cases of endangered languages with scant previously-available digital resources. The data used in the studies is from a language that UNESCO has categorized as *vulnerable*.

2.1.2 Documentation, description, and maintenance

Most work currently being done to address the problems of language endangerment falls under the titles of language documentation and description and/or language maintenance and revitalization. This section briefly discusses the aims of both language documentation and language description, following the discussion in Himmelmann (1998), Woodbury (2003), and Himmelmann (2008).

Under the traditional view, the desired end products of a language description consisted of a grammar, a dictionary, and perhaps a small collection of texts. In this view, according to Woodbury and Himmelmann, gathering a collection of texts was primarily a side effect of gathering the knowledge needed to produce a grammar and dictionary for the language. The usual case seems to have been for texts and field notes to remain unpublished and inaccessible beyond the individual researcher or research group. Working strictly according to this view may leave primary data inaccessible to other parties, making it difficult to verify analyses.

Himmelmann and Woodbury propose and define, respectively, a new field of documentary linguistics, related to but distinct from language descrip-

tion as traditionally construed. Both argue for a reconceptualization of the methodologies and desired end goals of a project to describe a previously undescribed or underdescribed language. Rather than focusing on static artifacts of textual analysis, language documentation should focus on collection of primary data. From Woodbury:

...direct representation of naturally occurring discourse is the primary project, while description and analysis are contingent, emergent byproducts which grow alongside primary documentation but are always changeable and parasitic on it.

This way of approaching language documentation crucially views language use as occurring in the context of the language community. In order to adequately document a language, one must document use of the language in different contexts, from different domains, and by different speakers. Examples drawn from texts to illustrate points in a grammar, e.g., should be clearly linked to the text from which they are taken, whether that be a recorded conversation, recorded storytelling, or via traditional elicitation.

To understand a language then requires an understanding both of the context of individual speech events and, more broadly, of the speech community and the range of strategies for language use practiced in the community. Correspondingly, to document that language requires capturing this range of language uses and annotating the collected texts with sufficient information that the ‘philologist 500 years from now’ (Woodbury, 2003) has what he or she needs to develop a linguistic analysis of the language.

This change in viewpoint brings with it a change in the ideal end products. Consequently, project goals are frequently defined in terms of hours of recorded text, recording quality, and other factors intended to produce documentation that is long-lasting and widely useful. Dobrin et al. (2009) essentially argue that the pendulum has swung too far, and that an excessive focus on quantifiable results can result in documentation projects that fail to meet the needs of the individual language community. They follow Himmelmann and Woodbury in emphasizing that the products and methodologies of any documentation effort need to be tailored to the specific community situation.

Here we make a quick note regarding language maintenance and revitalization. Language maintenance supports the transmission of a language to new speakers, working against language loss. Language documentation efforts can support this goal by including language-learning materials (or at least documentation that supports the development of pedagogical materials) in the set of end products. For example, texts with detailed annotation can be used by educators and language activists to create curriculum material for mother language education and to promote the survival of the language (Stiles, 1997; Malone, 2003; Biesele et al., 2009).

The view of language documentation espoused by Woodbury and Himmelmann has become the dominant way of approaching such projects. There seems to be relatively wide agreement that collections of texts with at least some level of analysis is a necessary, though perhaps not sufficient, component

of a successful language documentation effort.⁵

2.2 Language documentation process

One challenge for reusability of data from language documentation projects is the lack of consistency in the way that documentation is done. We refer in particular to variability in workflows, technologies, data formats, and annotation conventions. For the last decade or so, though, the field has been explicitly working on improving interoperability and consistency of annotation across language documentation projects.⁶ There is still no standard workflow,⁷ yet it *is* possible to outline, in broad strokes, a set of components to the documentation workflow.

The workflow we describe here represents the key tasks that must be accomplished to produce written records of recorded language use. Note that these are relevant regardless of the conceptualization of language documentation or the specific aims of a given documentation project, provided those aims include written representation of recorded speech.

This is intended as a high-level discussion of one possible documentation workflow. We do *not* aim to propose this as the best workflow. Rather, given

⁵Woodbury (2006) introduces the notion of ‘thick translation’ and outlines a strategy for documenting the meaning of a text at multiple levels and from a range of perspectives.

⁶Standards are one widely-discussed strategy for improving interoperability, but there are many issues and complications involved with development, implementation, and enforcement of standards. A discussion of these issues is beyond the scope of this thesis, but we point the interested reader to the following resources as a starting point: Bird and Simons (2003), Austin (2006), <http://emeld.org>, and <http://cyberling.elanguage.net>.

⁷Nor do we argue that there *should* be such a standard.

the wide range of approaches to documentation, as well as the multitude of available tools, formats, and standards, it is useful to narrow the space of inquiry to one model of the process. The workflow described here assumes that a linguist has already established a cooperative agreement in a community that wishes to participate in the documentation of its language. The workflow addresses only the parts of the process related to producing and managing written documentation of recorded language use.⁸ We do not address the development and management of a lexicon for the language, although the knowledge so obtained is crucial for carrying out many of the tasks described below. Finally, we use the term *the linguist* as a stand-in for any person or group of people working on a documentation project.

Orthography. For most endangered languages, there is little to no previous documentation. In many cases, the language has never been described at all. In order to produce written documentation of the language, then, the linguist must develop a system for writing the language. This is in fact a complex process requiring phonetic and phonological analysis as well as deciding whether and how to diagnose and indicate word boundaries in the language.

Transcription and translation. Once a writing system has been developed, recorded texts can be transcribed, producing the first, ‘raw’ record of

⁸The language documentation process, including important practical and ethical considerations, is discussed in much more detail in Gippert et al. (2006) and others.

the texts. Current best practice recommendations urge the linguist to align the transcriptions with their associated recorded media. Time-aligned transcription makes it possible to retrieve the original speech signal for any word, phrase, or clause of the written text. Texts are generally translated into a language of wider communication.

Morphological analysis. One aspect of linguistic analysis important for language description is an understanding of the language's morphological system. This includes determining the language's inventory of morphemes, knowing the meaning contributions and/or grammatical functions of individual morphemes and knowing how morphemes combine with stems and with each other. This process usually proceeds in tandem with analysis of the grammar of the language. For example, understanding how tense works in a language necessarily includes identifying and defining (if they exist) morphemes conveying temporal information. An important and non-trivial part of morphological analysis is determining a set of gloss labels to be used to represent morphemes' roles or contributions to meaning.

Text interlinearization and annotation. Interlinearization facilitates production of a written version of the text in which each clause or sentence is presented along with its translation and linguistic analysis. The latter appears in the form of morpheme-by-morpheme glosses of the sort discussed in the previous paragraph. The product of interlinearization — interlinear glossed

text, or IGT — is discussed at length in Chapter 3. The linguist’s task at this stage is to supply translations for stem morphemes and glosses for non-stem morphemes. Interlinearization and grammatical analysis are intertwined processes; it is very common for the linguist to develop new analyses based on examples encountered during interlinearization.

The current study specifically targets text interlinearization and annotation. We assume previous development of an orthography, basic understanding of the language’s morphology, and a set of pre-defined gloss labels, as dictated by the documentation project.

2.3 Annotation as ongoing analysis

One interesting characteristic of language documentation projects is the tentative nature of many analyses. Most annotation work involves analysis and re-analysis, often to refine a set of annotation conventions over time. In the language documentation context, annotation decisions can change dramatically as researchers come to understand the linguistics of the language.

Each of the four workflow components discussed in Section 2.2 incorporates and involves complex analysis of the phonological, morphological, syntactic, semantic, and perhaps pragmatic systems in the language being described. If these were independent systems, it might be possible to complete one stage of analysis and fix the conclusions for use in subsequent stages. Instead, these systems are interdependent in complex ways, and decisions at one stage of analysis may necessitate revision of earlier stages of analysis. At the same

time, some dependencies between stages do exist. One must, for example, decide on an orthography before any further written analysis can proceed.

Since analysis and annotation are often concurrent processes, with various levels or stages of analysis intertwined, language documentation creates a challenge for standard pipeline processing models in which each stage of a process must be completed before moving on to the next. More practically, the need to be able to make changes to earlier analyses creates complex data management problems for the working linguist. This is again because of the interdependent nature of the systems: a change in the orthography creates the need for transcriptions to be revisited and perhaps rewritten. That in turn necessitates examination of the morphological segmentations and glossing for affected portions of the text. Woodbury (2006) describes, in detail, how some of these interdependencies influenced an actual documentation project.

A second challenge is raised by the fact that analysis of a language may continue even once the documentation project has met its goals and completed its collection of annotated texts. Even fundamental parts of the grammatical analysis, such as the analysis of the case system or the nature of the verb constellation, can change over the course of time as more is learned about the language. In fact, creating and releasing such corpora *increases* the likelihood of revising earlier analyses as the data becomes available for study outside the immediate project group. Thus texts, and in particular text annotations, are ‘snapshots’ representing the state of the analysis at the time the texts were

finalized.⁹

The fluid nature of analyses in language documentation becomes a key factor in interpreting the results of the annotation experiments, as it is not always clear which of the conflicting analyses to take as the gold standard.

2.4 Computational linguistics and language documentation

One of the most extensive lines of related work aims to simplify and speed up the process of implementing computational grammars for a wide range of languages. Using the LKB (Copestake, 2001) system for implementing grammars in the HPSG formalism (Pollard and Sag, 1994), the Grammar Matrix project (Bender et al., 2002) aims to minimize development cost by appealing to crosslinguistic phenomena such as case or word order. Typologically-attested variations (e.g. SVO, SOV, etc. for word order) are encoded in general libraries, and these libraries are then used to jump-start grammar development (Bender and Flickinger, 2005).

A major advantage of working with an implemented grammar is the ability to efficiently test linguistic hypotheses against large amounts of data as well as against previous analyses through regression testing (Bender, 2008). A long term goal of the project is a system suitable for linguists documenting

⁹This is of course true of any corpus. The difference is that radical changes to grammatical analyses are far less likely for a language like English than for languages being described for the first time.

a new language. Early development of a computational grammar would allow the linguist to test analyses against the body of previously-collected data (Bender et al., 2004).

Another body of related work involves interlinear glossed text as the central data structure for language documentation. Bow et al. (2003) focus on modeling IGT, while Hughes et al. (2003), Schroeter and Thieberger (2006), and Palmer and Erk (2007) develop XML formats for working with IGT. This work, which is discussed in detail in Chapter 3, is theoretical work with direct relevance for tool development.

Recent work uses computational models and methods to support production of IGT. Moon et al. (2009) and Moon and Erk (2008) address the problem of identifying lemmas and segmenting word forms into their component parts. This work is briefly discussed in Chapter 3. Once word forms are segmented, the morphemes are labeled according to their contribution to the meaning of the sentence. Palmer et al. (2009) and Palmer et al. (submitted) use machine label suggestions and active learning to support the morpheme labeling part of the IGT production process. The latter work is discussed in chapters 5, 6, and 7.

Finally, it has been shown that IGT has potential as a rich resource that can be leveraged to perform other tasks. Drawing on the data collected in the Online Database of Interlinear Text (ODIN) (Lewis, 2006), Lewis and Xia (2008) use projected phrase structures (Xia and Lewis, 2007) to identify computationally relevant typological features such as case and word order.

Chapter 3

Interlinear glossed text

Currently there are many different strategies for and approaches to producing linguistically annotated texts from transcribed language data. Rather than considering each language case as a separate annotation problem, we focus on the widely-used interlinear glossed text (IGT) format. This chapter defines and discusses interlinear glossed text and proposes a new XML format for machine-readable representation and storage of interlinear glossed text.¹

3.1 Definition and general description

Interlinear glossed text (henceforth, IGT) is a format for coherent simultaneous presentation of multiple levels of linguistic analysis for a given piece of language data. Interlinear text translations interleave the source and target texts such that each line of the original text is aligned with a line of translated text.² The notion of interlinear glossed text in linguistics expands this format by inserting indications of different types of linguistic analysis between the original and the translated text, with each level of analysis generally

¹This chapter is based on and extends Palmer and Erk (2007).

²This is in contrast to, for example, translations of texts which present the original version on the left-hand page and the translation on the right-hand page.

represented by one line of text. (2) shows a line of Usphanteko text (TEXT) with its translation (TRANS) and an interlinear morphological segmentation of the Usphanteko words (MORPH). We also include an English translation.

- (2) TEXT: Kita' tinch'ab'ej laj inyolj iin
MORPH: kita' t-in-ch'abe-j laj in-yolj iin
TRANS: 'No le hablo en mi idioma.'

 ('I don't speak to him in my language.')

The multiple levels of analysis are generally aligned such that each word is visually associated (via vertical alignment) with its analyses. IGT is widely used in linguistics to present language examples, for example in reference grammars or journal articles, as it efficiently encodes complex multi-dimensional linguistic analyses. Some of the many other forms of analysis that may appear in IGT are phonetic transcriptions, phonemic transcriptions, word-by-word translations, part-of-speech tags, and prosodic information.

3.1.1 IGT for language documentation

Although there is no single common workflow for documenting endangered languages, and no single common set of desired outcomes, many documentation projects aim to produce textual corpora of transcribed and linguistically annotated speech.³ Frequently such corpora take the form of collections of texts in IGT format, with the IGT typically comprising at least four levels: the three shown in (2) plus a detailed morpheme-by-morpheme gloss.

³For further discussion see Section 2.2.

Examples (3) and (4) repeat the sentence in (2), adding the morpheme gloss line (GLOSS) and a part-of-speech tag line (POS). The Uspanteko IGT corpus uses this five-line format, shown below.⁴

- (3) TEXT: *Kita' tinch'ab'ej laj inyolj iin*
- (4) MORPH: *kita' t-in-ch'abe-j laj in-yolj iin*
 GLOSS: NEG INC-E1S-hablar-SC PREP E1S-idioma yo
 POS: PART TAM-PERS-VT-SUF PREP PERS-S PRON
 TRANS: ‘No le hablo en mi idioma.’
 (‘I don’t speak to him in my language.’)

Note that the GLOSS line includes two different types of labels. Labels for non-stem morphemes indicate the grammatical function of the morpheme (e.g. **NEG** for *kita'*), and stem morphemes are labeled with translations into what Woodbury (2003) refers to as the ‘language of wider communication’ (e.g. *hablar* ‘to speak, to speak to’ for *ch'abe*).

The POS line contains a mix of part-of-speech tags and broader class labels of various types, such as **TAM** for morphemes conveying information related to tense, aspect, or mood. These two types correspond to the two types of labels on the GLOSS line: stems are labeled with part-of-speech tags, and grammatical morphemes are marked with broader class labels. To continue our example, the stem *ch'abe* is labeled as a transitive verb (**VT**), and the

⁴KEY: E1S=singular first person ergative, INC=incompletive, PART=particle, PREP=preposition, PRON=pronoun, NEG=negation, S=sustantivo (noun), SC=category suffix, SUF=suffix, TAM=tense/aspect/mood, VT=transitive verb

stand-alone morpheme *kita*’ is marked as belonging to a very general category of particles (PART).

3.1.2 Variability in IGT

One advantage of IGT as a means for encoding linguistic analysis of transcribed texts is precisely its flexibility; the format can be easily adapted to meet the varying requirements of different projects. In this section we discuss some of the ways in which IGT formats may vary from each other.⁵

A first, obvious manner of variation is at a high level, in the data structure itself. A broad survey of formats for interlinear texts (Bow et al., 2003) found wide variation in both the number of annotation tiers, the type of analysis found in each tier, and the level of granularity of analysis in each tier. This aspect of the IGT format is generally determined by the goals, research or otherwise, of the individual or team coordinating production of IGT. In other words, the number, type, and granularity of tiers are defined at the project level.

Second, there is wide variation in the types of labels used for any given type of annotation tier. In the language documentation context, this variation is most salient with reference to the GLOSS line, and in particular to the gloss labels for non-stem morphemes. The task of labeling morphemes according to

⁵Examples of interlinear text from an extremely wide range of languages (and in many different formats) are available from the Online Database of Interlinear Text. <http://www.csufresno.edu/odin>

their grammatical function is deeply intertwined with the process of linguistic analysis itself and thus tends to inherit any theoretical predispositions of those performing the analysis. The situation is further complicated by the fact that one term (for example ‘nominative’) can mean different things to different communities of linguists. Thus, the particular label set used for annotating a corpus is often based on linguistic or theoretical traditions and fine-tuned by the individual project. One line of work addressing this issue seeks not to impose a common set of labels but rather to provide an ontology of linguistic terminology to which individual label sets can be mapped (Farrar, 2007; Farrar and Langendoen, 2003).

In addition to using different sets of labels, projects tend to use different conventions for combining labels. For example, on the GLOSS tier of annotation, morphological segmentations of words may be separated by hyphens, dots, @ signs, or other symbols. The Leipzig Glossing Rules⁶ are a recent movement toward a standardized “syntax” and “semantics” for inter-linear glosses. The Leipzig Rules are proposed not as a fixed standard but rather as a set of conventions which, for the most part, simply reflect and codify what is already common practice in the linguistics community. It should be noted that the Rules reflect common practice in the *presentation* of IGT.⁷ For machine-readability, we need consistency not in presentation of IGT but in the underlying *structural representation* of the data.

⁶<http://www.eva.mpg.de/lingua/resources/glossing-rules.php>

⁷Hughes et al. (2003) also discusses variation in presentational factors.

Modeling interlinear text. Building on the analysis in Bow et al. (2003) of different IGT formats used in the literature, Hughes et al. (2003) propose a four-level hierarchical model for representing interlinear text; we refer to this as the **BHB model**. The four levels encode elements common to most instances of IGT: *text*, *phrase*, *word*, and *morpheme*. One *text* may consist of several individual *phrases*. A *phrase* consists of one or more *words*, each of which consists of one or more *morphemes*. Referring back to Section 3.1.1, (3) shows a single phrase (or a one-phrase text). The four annotation tiers in (4) are situated at different levels in the hierarchy: the first three tiers are situated at the morpheme level, and the final tier, the translation, is again situated at the phrase level, like the original text in (3).

3.1.3 Machine-readable IGT

One important aim of language documentation is to record and preserve language data in ways that will be accessible and useful to different users (for example, native speakers, community language teachers, or linguists of various stripes) both now and in the future. For the purposes of electronic archiving and presentation, as well as for computational analysis and support, a machine-readable version of the corpus is needed.

The documentary linguistics community has for some time been engaged in discussion and development of technologies and best practice recommendations for long-term digital storage of language data. For example, the EMELD project and its series of workshops produced a large volume of work

on this topic as well as a website with links to resources and recommendations for best practices.^{8,9} Another important resource is an extensive discussion in Bird and Simons (2003) of requirements for achieving interoperability and portability in language documentation.

Interoperability and adherence to standards or best practices are largely concerned with consistency and compatibility. A distinction can be drawn between compatibility of formats across documentation projects—**external consistency**—and **internal consistency**, or regularity of structure, annotations, and analysis within one individual project. Chapter 4 addresses our work to achieve internal consistency in the Uspanteko corpus. Issues of external consistency are discussed in the next section.

3.2 The IGT-XML format

Producing a machine-readable version of the corpus involves a number of choices about formats and standardization. In the absence of a machine-readable format for IGT with the flexibility needed for our research, we develop a new format called **IGT-XML**. This section describes the format, discusses requirements for machine-readable IGT, and shows how our format satisfies those requirements.

⁸<http://emeld.org>

⁹This same concern is of growing interest to linguistics at large. See for example outcomes from the 2009 Cyberling workshop, which was dedicated to making progress on a cyber-infrastructure for linguistics, with a particular focus on data publication, citation, storage and sharing. <http://cyberling.elanguage.net>

We choose XML as the framework for our format because it supports the goals of **archiving** and **portability**. In addition, XML is the current best practice recommendation for ensuring both format interoperability and preservation of data (Bird and Simons, 2003).

We build on the **BHB model**, the first to propose using the IGT structure directly as a basis for an XML format. However, our model has a more loosely coupled and flexible representation of different annotation layers, to accommodate (a) selective manual reannotation of individual layers, and (b) the (semi-)automatic extension of annotation, without the format posing an *a priori* restriction on the annotation levels that can be added.

3.2.1 Basics of IGT-XML

The IGT-XML format we use for the Uspanteko data contains five main components, illustrated in Figure 3.1:

- a **plaintext** component comprising phrases as well as the individual words making up each phrase, encased in the `<phrases>` XML element,
- a **translation** component comprising sentence-level glosses, encased in the `<translations>` XML element,
- a **morpheme** component giving a morphological analysis of the source text, encased in the `<morphemes>` XML element,
- a **gloss** component including morpheme-level glosses, encased in the `<gloss>` XML element, and

- a **part-of-speech** component comprising part-of-speech tags and broader category labels (see Section 3.1.1), encased in the `<pos>` XML element.

Annotation is grouped into blocks in a modular fashion, each block representing an annotation layer. Each block is internally structured by phrases, and one level below phrases, but there is no deeper embedding. In particular, morphemes are **not** embedded within words, such that additional layers of annotation at the word and the morpheme levels can be added modularly without interfering with each other.

Modularity. Further annotation layers can be added by extending the format by additional components beyond these five, which describe the core four levels of interlinear text and one additional project-defined level.

For example, IGT-XML is easily extended with **metadata** for each texts.¹⁰ In fact, because metadata is a separate level of annotation, we could also encode phrase-level metadata, if for example we wanted to indicate change of speaker in recorded conversation.

The **flexibility** of the format is useful for language documentation projects, as linguistic analysis of the language data is often tentative and subject to change.¹¹ Flexibility is essential for the experiments in semi-automatic

¹⁰The figure suggests use of the OLAC metadata standard, which is oriented toward archiving of language data (<http://www.language-archives.org>). This is one of many possible approaches to handling metadata.

¹¹see Chapter 2 for further discussion.

analysis presented in this thesis. We require a format that is flexible as to which layers of annotation are present and in which order they are added. It should also allow us to store, side by side, labels created by a human annotator and machine-created labels for the same layer of annotation.

Flexibility requirements can be met by having different layers of annotation that are not coupled tightly, such that individual layers can be exchanged without affecting others. Stand-off annotation formats achieve this, by having independent annotation layers whose only link is the reference to a common plain text. We adopt a moderately stand-off solution, in which different annotation layers share a single file and are grouped sentence-wise, but are still linked via references to a common source text.

3.2.2 Details of the format

Within the `<phrases>` block, each individual phrase is encased in a `<phrase>` element, which includes the raw text of the phrase within the `<plaintext>` element as well as each individual word of the text in a `<word>` element. Each `<phrase>` and each `<word>` has a globally unique ID, assigned in an `id` element, which allows parts of the annotation to point to other phrases or words.

The `source_layer` and `source_id` attributes preserve identifying labels assigned by the author of the original texts. In the case of representing Toolbox data, the value of `source_layer` is the user-selected code for each layer of annotation, the `ref` line in Figure 3.2. Values for `source_id` are the user-assigned

```

<text id="T1" lg="usp" source_id="trtex068" title="example">
<metadata idref="T1">
<!-- OLAC metadata elements -->
</metadata>
<phrases>
  <phrase ph_id="T1_P1">
    <plaintext>xelch li+</plaintext>
    <word text="xelch" wd_id="T1_P1_W1"/>
    <word text="li" wd_id="T1_P1_W2"/>
  </phrase>
</phrases>
<morphemes>
  <phrase phrase_ref="T1_P1">
    <morph morph_id="T1_P1_W1_M1" text="x-"/>
    <morph morph_id="T1_P1_W1_M2" text="el"/>
    <morph morph_id="T1_P1_W1_M3" text="-ch"/>
    <morph morph_id="T1_P1_W2_M1" text="li"/>
  </phrase>
</morphemes>
<gloss>
  <phrase phrase_ref="T1_P1">
    <gls idref="T1_P1_W1_M1" text="COM"/>
    <gls idref="T1_P1_W1_M2" text="salir"/>
    <gls idref="T1_P1_W1_M3" text="DIR"/>
    <gls idref="T1_P1_W2_M1" text="DEM"/>
  </phrase>
</gloss>
<pos>
  <phrase phrase_ref="T1_P1">
    <pos idref="T1_P1_W1_M1" text="TAM"/>
    <pos idref="T1_P1_W1_M2" text="VI"/>
    <pos idref="T1_P1_W1_M3" text="DIR"/>
    <pos idref="T1_P1_W2_M1" text="PART"/>
  </phrase>
</pos>
<translations>
  <phrase idref="T1_P1">
    <trans id="T1_P1_Tr1" lg="esp" text="salio entonces"/>
    <trans id="T1_P1_Tr2" lg="en" text="then he left"/>
  </phrase>
</translations>
</text>

```

Figure 3.1: IGT-XML: Uspanteko clause.

reference numbers for texts or individual phrases or clauses of text.

The morphemes in the `<morphemes>` block are again organized by `<phrase>`. Each `<phrase>` in the `<morphemes>` block refers to the corresponding phrase in the `<phrases>` block by that phrase's unique ID.

Each individual morpheme, represented by a `<morph>` element, refers to the `<word>` of which it is a part, via that word's unique ID, specified in the *idref* attribute. The linear order of morphemes belonging to the same word is reflected in the order in which `<morph>` elements appear, as well as in the running *id* of the morphemes. Morphemes have *id* attributes of their own such that further annotation levels can refer to the morphological segmentation of the source text.

All glosses are collected in the `<gloss>` block. Again, they are organized by `<phrase>`, linked to the original phrases by *idref* attributes. The glosses within each `<phrase>` refer to individual morphemes, hence their *idref* attributes point to *id* attributes of the `<morphemes>` block. The POS line labels are organized in the same manner as the glosses.

3.3 Related work

This section discusses related work on XML formats for interlinear text and tools for creating and working with IGT. We also discuss work on automating production of IGT.

3.3.1 XML formats for IGT

We are aware of two other XML formats that are specific to interlinear text but *not* tied to any particular language.

The BHB XML format. The XML representation presented in Hughes et al. (2003) articulates the four nested levels of structure of the BHB model discussed in Section 3.1. The BHB format directly expresses the hierarchy of annotation levels in a nested XML structure, in which, for example, XML elements representing morphemes are embedded in XML elements representing the corresponding words. The model maintains the link between the source text morpheme and the morpheme-level gloss annotation by embedding both items within the morpheme level of structure and using a type attribute to distinguish between the two.

This representation is less flexible than IGT-XML because it is not modular. To add an additional annotation layer at the morpheme level, one would need to access and change the representation of each morpheme of each word of each phrase. In this way, the BHB XML format is not well-suited for present purposes.

A more flexible XML format for IGT, largely tailored to flexibility in presentation, is introduced in Schroeter and Thieberger (2006).

```

\ref trtex068Usp04_286
\t kita' tinch'ab'ej      laj inyolj      iin+
\m kita' t-  in-  ch'abe -j laj in-  yolj  iin
\g NEG  INC-  E1S-  hablar -SC PREP E1S-  idioma yo
\s PART  TAM-  PERS-VT      -SC PREP PERS-S      PRON
\l no le hablo en mi idioma

```

Figure 3.2: Toolbox output: Uspanteko clause

3.3.2 Tools for IGT

The idiosyncratic nature of language documentation projects makes it very difficult to develop general-purpose tools for interlinear glossing of texts and production of IGT. Some existing tools are discussed below.

Shoobox/Toolbox¹² (*Toolbox* in following text) is a system that is widely used in documentary linguistics for storing and managing language data. It provides facilities for lexicon management as well as text interlinearization. The custom whitespace delimited format generated by Toolbox is perhaps the most widespread format for digital representation of IGT, but the format makes normalization into a structured representation particularly challenging. Figure 3.2 repeats (4), this time displayed in the standard Toolbox output format. It should be noted that the whitespace is part of the data itself, not something inserted for presentational purposes.

Another challenge is that the glossaries, grammatical markers and segmentations are defined at the individual project level, and it can be challenging

¹²Freeware downloadable from http://www.sil.org/computing/catalog/show_software.asp?id=79.

for an incoming linguist to learn how these are defined. The Fieldworks Language Explorer (FLEX) is a newer, open-source system for lexicon and text management, as well as glossing and interlinearization.

Many projects use (mostly proprietary) general-purpose word processing, spreadsheet, and database software for data management and text interlinearization. The same problems with project definitions arise under this approach.

The **Interlinear Text Editor** (Lowe et al., 2004)¹³ provides a straightforward user interface for transcription and glossing of texts. The tool works on a default underlying data structure but allows user modification to work with other formats. ITE also builds a minimal lexicon as transcription proceeds, but it does not offer the extended lexicon management capabilities provided by Toolbox.

Finally, **TypeCraft**¹⁴ is a free, wiki-based platform for collaborative annotation and sharing of interlinear text. Unlike Toolbox and ITE, TypeCraft restricts annotators to a predetermined set of gloss labels. TypeCraft also assumes a three-way distinction of gloss types: translational (as on STGLOSS tier in Table 3.1), functional (MGLOSS), and part-of-speech (STPOS) glosses (Beermann and Mihaylov, 2009).

Many individual projects have developed excellent resources for anno-

¹³ITE is open-source, available from http://michel.jacobson.free.fr/ITE/index_en.html.

¹⁴<http://www.typecraft.org>

tation, reuse, and web-based dissemination of their *own* data; what is lacking is an up-to-date, thoroughly general, and user-friendly system.

Converting other formats to IGT-XML. Each system managing IGT data has different output formats, requiring different techniques for transforming the data to IGT-XML. Even when projects use the same tool, for example Toolbox, the abundance of project-defined settings means that Toolbox-produced IGT from one project can differ wildly from Toolbox-produced IGT from a second project. Thus each format conversion needs some amount of customization. Chapter 4 describes clean-up of the (Toolbox-produced) Us-panteko data and its conversion to IGT-XML.

3.3.3 Varied practices for IGT production

To better understand data management needs and current practices in language documentation, we conducted an informal survey of linguists in the University of Texas Linguistics Department who were working on documentation projects.¹⁵ The five projects surveyed were at different stages in the process, ranging from very early stages to the late stage of having a nearly complete text corpus and reference grammar of the language. The main finding of the survey, which focused on aspects of documentation relating to production of IGT, is that approaches vary widely.

Only two of the five projects surveyed had digitized texts with full

¹⁵Taesun Moon, p.c.

IGT: transcription, translation, and morpheme glossing. Two projects had partial IGT for their texts: transcription and translation, but no morpheme-level glossing. The remaining project had no full texts to work with, but rather was at the early stage of eliciting individual lexical items. There was also wide variation in software used for transcription and/or glossing. Two projects used Toolbox,¹⁶ two used ELAN¹⁷ (one of those in conjunction with Microsoft Excel), and the fifth used a combination of Microsoft Word and Microsoft Access. Each of these software packages uses its own underlying data structures for storing and representing (partial or complete) IGT.

3.3.4 Automated production of IGT

Producing IGT automatically has the potential to reduce the amount of human time required, thereby speeding up the language documentation process. However, there are many open questions regarding the degree of automation to be sought, and how to automate in a manner that works well with linguists and users helping to develop IGT resources.

It is helpful to first consider the subparts of IGT production individually. For the following discussion we assume that existing transcriptions and translations of recorded speech in some language are available at the start of automation.

The first subpart is to perform morphological segmentation, break-

¹⁶<http://www.sil.org/computing/shoebox>

¹⁷<http://www.lat-mpi.eu/tools/elan>

MORPH:	kita'	t-	in-	<i>ch'abe</i>	-j	laj	in-	<i>yolj</i>	<i>iin</i>
MGLOSS:	NEG	INC-	EIS-		-SC	PREP	EIS-		
STGLOSS:				<i>hablar</i>				<i>idioma</i>	<i>yo</i>
STPOS:				<i>VT</i>				<i>S</i>	<i>PRON</i>
MPOS:	PART	TAM-	PERS-		-SUF	PREP	PERS-		

Table 3.1: Decomposition of GLOSS and POS tiers of IGT.

ing each word into its component morphemes. Because of data scarcity, approaches that learn morphology from unlabeled data are especially interesting. Unsupervised morphology acquisition is an active area of research, and here we mention just three recent works. Moon et al. (2009) use document boundaries to constrain stem generation and clustering. Minimum description length models are used by Creutz and Lagus (2007) and Goldsmith (2001). A more thorough review of relevant work appears in Moon et al. (2009). Much remains to be done before morphological induction achieves maturity in the context of reliable automation.

The next subparts—producing the GLOSS and POS annotation tiers—combine translation, tagging, and simple mapping tasks. To see this more clearly, we appeal to the distinction between stem and non-stem morphemes. For purposes of illustration, Table 3.1 divides each of the tiers into two sub-tiers, one for stem morphemes and one for non-stem morphemes. All stem morphemes and their labels appear in italics.

Stem morphemes are labeled on the gloss line (STGLOSS) with a translation of the stem’s meaning into the research or reference language. On the STPOS line, the stem is represented by its part-of-speech tag. These two

labels (e.g. *hablar-VT* for the stem *ch'abe*) constitute a minimal lexical entry for the stem.¹⁸

Non-stem morphemes are labeled on the gloss line (MGLOSS) with functional/grammatical glosses. Producing these labels can be viewed as a part-of-speech tagging task with an expanded tagset; capturing the meaning contributions of all morphemes requires a larger set of labels than the standard part-of-speech tagset. Finally, the labels on the MPOS line are determined by mapping from a morpheme gloss (e.g. *INC* for incomplete aspect) to its higher-level category (in our example, *TAM*, for tense-aspect-mood).

There is potential to automate IGT production, but it is important to do so in a way that maintains quality, consistency, and utility for future generations of linguists who will rely on data produced in such a manner.

¹⁸This particular subtask, which is much simpler than full interlinear glossing, may be suitable for annotators who know the language but have minimal background in linguistics.

Chapter 4

Data and data preparation

This chapter describes the Uspanteko corpus used for the studies presented in chapters 5 and 6 and discusses preprocessing required before the data could be used for training and evaluation of machine learners. We argue for an iterative, **cooperative** procedure for improving the internal consistency and reusability of corpora from language documentation projects. In doing so, we address a number of error types frequently found in interlinear texts.¹

As discussed in Section 3.1.3, for data to be widely reusable, curation and annotation of the data must consider both external and internal consistency. The focus in this chapter is on **project-internal** consistency and its relation to **machine reusability** of project data, and specifically of interlinear glossed text (IGT).

4.1 The corpus: Uspanteko texts

This section describes the IGT corpus used in our experiments. The corpus is a set of texts (Can et al., 2007a) in the Mayan language Uspanteko.

¹This chapter incorporates and extends material from Palmer et al. (2009) and Palmer et al. (submitted).

Uspanteko is a member of the K'ichee' branch of the Mayan language family and is spoken by approximately 1320 people, primarily in the Quiché Department in west-central Guatemala (Richards, 2003). The texts were collected, transcribed, translated into Spanish, interlinearized, and glossed as part of a Mayan language documentation project carried out by La Asociación Oxlajuuj Keej Maya' Ajtz'iib' (OKMA).² The OKMA institution is based in Antigua, Guatemala and is dedicated to the study and documentation of indigenous languages of Guatemala. The Uspanteko texts are currently accessible via the Archive of Indigenous Languages of Latin America (AILLA).³

The portion of the Uspanteko corpus we use contains 67 texts with various degrees of annotation. All 67 texts have been transcribed, several translated but not glossed, and 32 of the texts have full transcriptions, translations, morphological segmentation, and glossing.⁴ Of the 284,455 total words of text in the corpus, 74,298 are segmented and glossed. The full IGT texts are represented in the Toolbox output format, as shown in Figure 3.2. The transcribed and translated texts are like the Uspanteko sample shown below (text 068, clauses 283-287):

- (5) a. Uspanteko: *Non li in yolow rk'il kita' tinch'ab'ex laj inyolj iin, si no ke laj yolj jqaaj tinch'ab'ej i non qe li xk'am rib' chuwe, non qe li lajori non li iin yolow rk'ilaq.*

²<http://www.okma.org>

³<http://www.ailla.utexas.org>

⁴The set of texts available at AILLA varies somewhat from the set we used.

- b. Spanish: *Sólo así yo aprendí con él. No le hablé en mi idioma. Sino que en el idioma su papá le habló. Y sólo así me fui acostumbrando. Sólo así ahora yo platico con ellos.*
- c. English: *And so I learned with him. I did not speak to him in my language [K'ichee']. But his father spoke to him in HIS language [Uspanteko]. That's how I got used to it, and so now I speak with them.*

The glossed texts are of four different genres. Five texts are oral histories, usually having to do with the history of the village and the community, and another five are personal experience texts describing events from the lives of individual people in the community. One text is a recipe, and another is an advice text in which a speaker describes better ways for the community to protect the environment. The remaining twenty texts are stories, primarily folk stories and children's stories. This is a small dataset by current standards in computational linguistics, but it is rather large for a documentation project.

4.2 Data clean-up and conversion

The examples of Uspanteko seen in the previous chapter all were perfectly segmented, perfectly labeled, and perfectly aligned. Here we repeat (4) from Chapter 3 as (7), slightly modified to show hyphenation conventions.

- (6) TEXT: Kita' tinch'ab'ej laj inyolj iin
- (7) MORPH: kita' t- in- ch'abe -j laj in- yolj iin
 GLOSS: NEG INC- EIS- hablar -SC PREP EIS- idioma yo
 POS: PART TAM- PERS- VT -SUF PREP PERS- S PRON
 TRANS: 'No le hablo en mi idioma.'
 ('I don't speak to him in my language.')

Each morpheme is assigned precisely one label, and the crucial MORPH and GLOSS tiers each contain the same number of elements. On these two tiers and the POS tier, hyphenation conventions consistently indicate stem and affix status. Affixes take hyphens, pointing in the direction of the stem, and stems remain unhyphenated.

As is the case with many corpora, the original Uspanteko data contained a number of inconsistencies and incomplete annotations. Small inconsistencies in labeling and hyphenation occur for even the most thorough annotators; it is easy to make such errors when the annotation tool has no way of enforcing consistency of annotation. In addition, the data is presented in the loose, space-delimited Toolbox format (see Section 3.3.2).

Consistency in labeling and alignment is essential for IGT data to be smoothly handled in our experiments. The machine learner must be able to extract the information encoded in the IGT representation accurately and systematically. For the morpheme labeling task, it is crucial that the links between annotation tiers are maintained, so that each morpheme is properly linked to its associated label. Maintaining these links also involves ensuring that the alignment between tiers is preserved. Finally, the labels used must be

consistent and free of typographical errors in order for the learner to accurately model the contexts in which a given label occurs. To enable reliable extraction of morpheme segmentation and glosses for measuring the performance of our models, it was necessary to clean up such annotation gaps and errors.

4.2.1 Cooperative data clean-up

This section offers a detailed discussion of our data clean-up process and argues for a cooperative approach to data clean-up in which a language expert (**LgExp**) and a computational linguist (**CompLx**) work side by side. We discuss different types of inconsistencies found in the data and what type of expertise is needed to resolve them.

Language documentation corpora in general share several common sources of labeling inconsistency. First, textual data from endangered languages, many of which have never been written down before, tend to require more preprocessing than text that was written down to start with. One reason for this is that the orthography and the grammatical analyses that form the basis of the associated writing system are often in a state of flux during the documentation process. In addition, the vast majority of documentary data are from transcribed spoken texts, often spontaneous speech or story-telling, with the usual dysfluencies, false starts, repetitions, and incomplete sentences. The annotations of the transcriptions inherit this messiness. Finally, text interlinearization is sometimes done by multiple annotators with varying levels of knowledge and/or expertise, both language-specific and pertaining to linguistic

analysis.

One type of error is inconsistency in labeling. When the inconsistency involves typographical variation (e.g. **PST** and **pst** both used to indicate a past tense morpheme), no language-specific expertise is required to recognize and resolve the inconsistencies. Once such errors have been identified, a linguist (computational or otherwise) with basic programming skills and a decent command of regular expressions and a scripting language can correct them automatically and very quickly.

A different sort of labeling inconsistency arises from inconsistency in analysis: one annotator may view the morpheme as indicating past tense and choose the **PST** label, while another may interpret the morpheme as conveying completive aspect and mark it with **COM**. Analytic inconsistency can even occur for a single annotator, if his or her analysis of tense and aspect in the language changes over the course of annotation. Inconsistencies of this type require adjudication by a linguist who knows the language as well as the current ‘correct’ analysis of the particular linguistic phenomenon.

In order to perform corpus clean-up quickly and accurately, we devised a cooperative approach, combining the two skill sets of language-specific expertise and computational text processing. It is highly efficient for the two to work side by side in an iterative error identification and correction process.

The particular computational methods required, as well as the details of performing automatic identification and correction, will be specific to the

individual documentation project. In our case, we applied standard scripting, concordancing, and search-and-replace techniques, including heavy use of regular expressions. We aimed for the simplest script or code possible to zoom in on potential errors with no manual search of the corpus, only manual adjudication of possible errors.

The cooperative corpus clean-up approach takes advantage of complementary knowledge and skill sets to facilitate efficient error correction with minimal need for additional training. It is a rapid and targeted way to improve the consistency of a corpus as well as its suitability for use as training material for machine learners or for other forms of automated processing or analysis.

4.2.2 Error types and correction methods

In this section we discuss the main types of errors and inconsistencies we identified and corrected in the Uspanteko data. While the details of such errors of course vary according to the project, the classes of errors discussed below are quite general and are found in many IGT corpora.

Grouping of annotation tiers. For each clause of interlinear glossed text, there should be a TEXT tier, a MORPH tier, a GLOSS tier, a POS tier, and a TRANS tier. In a whitespace-delimited format, grouping of annotation tiers is often indicated by inserting a blank line between each clause-level grouping, and errors in this grouping (e.g. extra blank lines *between* related annotation

tiers, or absence of a blank line between tiers for two different clauses) are easy for a human to diagnose but tedious to correct. At the same time, getting this basic grouping right is essential for any subsequent automated processing. We used a simple script to produce a list of suspect clauses requiring attention to better target manual review.

Label consistency. Because most systems used for IGT production do not restrict the set of possible labels, inconsistent labels occur reasonably frequently in language documentation annotation. Some errors are typographical (e.g. labeling a future-tense morpheme with FIT instead of FUT). Others stem from a lack of agreement on conventions for capitalization and punctuation of labels; in our case the label for third-person singular ergative marking showed up in all the following variations: E3S., E3s., e3s., E3S, E3s, e3s. Straightforward UNIX command line utilities allowed us to quickly build a list of all tags in the corpus, which at its largest contained over 200 different tags. The list was adjudicated by the **CompLx** with assistance on several points from the **LgExp**, and a final list of 69 possible labels was agreed upon. Simple search-and-replace functions were used to correct the errors. Note that this use of search-and-replace, together with concordancing, could also be very useful to help the linguist back-propagate changes in analysis, orthography, or labeling conventions that occur *during* annotation.

Consistency of hyphenation. A challenge for representing IGT in a machine-readable format, especially starting from a minimally-structured representa-

CORRECT	x-	e1	-ch
TOO MUCH	x-	-e1-	-ch
TOO LITTLE	x	e1	ch
MIXED	x	-e1	-ch

Table 4.1: Hyphenation possibilities for a three morpheme word form.

tion, is to treat each morpheme as an individual token while preserving the links between words on one line and morphemes on the next. We use hyphenation conventions to indicate groups of morphemes associated with a common word: prefixes get a right-side hyphen, suffixes get a left-side hyphen, and stems remain bare. Hyphenation patterns in the original texts varied a great deal. For example, the word form `x-e1-ch` (COM-salir-DIR) could appear with many different hyphenations, some of which are shown in Table 4.1. To address hyphenation errors, we built on the previous step of normalizing gloss labels: stem morpheme labels are consistently all lower-case, and labels for non-stem morphemes are all upper-case. This automatic morpheme type identification is combined with targeted manual correction.

Alignment of annotation tiers. It is also crucial to properly maintain links between source text morphemes and the gloss labels assigned to them. Specifically, the MORPH, GLOSS, and POS lines must all have the same number of items. We again used scripting procedures to identify such errors, but resolving them required manual review. Some misalignments come from bad segmentation, as in (8) and (9). Here the number of elements in the MORPH line does not match the number of elements in the GLOSS line. The problem

in this case is a misanalysis of *yolow*: it should be broken into two morphemes (*yol-ow*) and glossed *platicar-AP*.⁵

- (8) TEXT: Non li in yolow rk'il
- (9) MORPH: Non li in yolow r-k'il
GLOSS: DEM DEM yo platicar AP E3s.-SR
POS: DEM DEM PRON VI SUF PERS SREL
TRANS: 'Sólo así yo aprendí con él.'

Other alignment errors come from gaps in annotation. Even among the 32 glossed texts, not all are fully annotated. Most include occasional instances of partial annotation at the clause, word, or morpheme level. To maintain tier-to-tier alignment, each morpheme needs *some* label on each tier, even if only to indicate that the label is unknown. Some missing labels were filled in by the **LgExp**. Others were filled with the placeholder label ('???'). The version of the corpus used in the experiments includes 468 known morphemes labeled with '???'.⁶

Conversion to IGT-XML. Finally, once word-to-morpheme and morpheme-to-gloss alignment problems are resolved, the cleaned annotations can be converted into IGT-XML (Chapter 3). To do this we used a combination of the Shoebox/Toolbox interfaces provided in the Natural Language Toolkit

⁵KEY: AP=antipassive, DEM=demonstrative, E3S=singular third person ergative, PERS=person marking, SR/SREL=relational noun, VI=intransitive verb

⁶An additional 734 instances of the '???' label appear in cases of untranscribed morphemes. These are cases where the segmentation in the original corpus indicates the existence of a morpheme without indicating its identity.

(Robinson et al., 2007) and Python code written specifically for handling the Uspanteko data. The conversion process is straightforward, but the many preprocessing steps described here are crucial for making it so.

It is worth noting that documentary linguistics projects can benefit greatly from performing a semi-automated clean-up process and converting formats in this manner. The resulting corpus should be useful for future corpus and computational studies. In addition, the automated clean-up process itself can be fruitful for linguistic analysis. On some occasions, the scripts uncovered discrepancies in analysis or interesting error patterns that led to deeper analysis and new insights into some aspect of the language.

4.3 Target representation

In production of IGT, two key tasks are word segmentation (determination of stems and affixes) and glossing each segment. As discussed in Section 3.3.4 and illustrated in both Table 3.1 and Table 4.2, stem morphemes and non-stem morphemes each get a different type of gloss. The gloss of a stem is usually its translation (STGLOSS) whereas the gloss of a non-stem morpheme is a grammatical label (MGLOSS). The additional word-class line (STPOS) provides part-of-speech information for the stems, such as **VI** for *platicar* “to talk, to chat”.

The target representation for the semi-automated annotation studies (Chapter 6) combines STPOS labels and MGLOSS labels on a single line. The result is an additional tier, labeled COMBO in (11). This representa-

MORPH:	Non	li	in	<i>yol</i>	-ow	r-	k'il
MGLOSS:	DEM	DEM			-AP	E3S-	SR
STGLOSS:			<i>yo</i>	<i>platicar</i>			
STPOS:			<i>PRON</i>	<i>VI</i>			
MPOS:	DEM	DEM			-SUF	PERS-	SREL

Table 4.2: Repeat: Decomposition of GLOSS and POS tiers of IGT.

tion simplifies the automated glossing process by obscuring the need to learn translations for all stems occurring in the texts. In an actual documentation project, of course, *both* STGLOSS and STPOS labels would be provided as part of the glossing process. However, stem translation is beyond the scope of this dissertation, so we focus on predicting a refined set of gloss/POS labels. Example (10) repeats the clause in (8), adding this new combined tier. Stem labels are given in bold text, and affix labels in plain text.

(10) TEXT: Non li in yolow rk'il

(11) MORPH: Non li in yol-ow r-k'il
COMBO: DEM DEM **PRON VI**-AP E3S-SR

TRANS: 'Sólo así yo aprendí con él.'

A simple procedure was used to create the new tier. For each morpheme, if a gloss label (such as DEM or E3S) appears on the gloss line (GLOSS), we select that label. If what appears is a stem translation, we instead select the part-of-speech label from the next tier down (POS).

In the entire corpus, sixty-nine different labels appear in this combined tier. Table 4.3 shows the five most common part-of-speech labels (left) and the

S	noun	7167	E3S	sg.3rd ergative	3433
ADV	adverb	6646	INC	incompletive	2835
VT	trans. verb	5122	COM	completive	2586
VI	intrans. verb	3638	PL	plural	1905
PART	particle	3443	SREL	relational noun	1881

Table 4.3: Most common labels and their frequencies.

five most common gloss labels (right). The most common label, **S**, accounts for 11.3% of the tokens in the corpus.

Chapter 5

Active learning with simulated annotation

Active learning (AL) is an approach to machine learning which focuses on reducing the annotation effort required to develop labeled data sets, often with the goal of using the labeled data as training material. Alternately, the goal can be stated in terms of using annotation resources as effectively as possible in order to improve the performance of the learned model.

In natural language processing, the resources available in the prototypical AL situation are a small amount of labeled data (sometimes very small), a much larger amount of unlabeled data, and annotation resources sufficient to label some but not all of the unlabeled data. Active learning is a natural choice for language documentation because documentation projects tend to have a similar balance of labeled and unlabeled data, as well as having limited resources to devote to further annotation. Further discussion of AL is the focus of Section 5.1 of this chapter.

The studies described in the remainder of the chapter do active learning with simulated annotation for part-of-speech tagging. These studies are preliminaries to the live annotation AL experiments discussed in Chapters 6 and 7. The simulated studies verify that our system produces the expected

results for simulated active learning. In addition, they establish the system's performance on the Uspanteko data.¹

5.1 Active learning

Active learning attempts to maximize the impact of annotation effort by identifying informative examples for the human to annotate. Examples which present some novelty are likely to help a machine learner improve its performance more quickly than are frequently-occurring examples.

In one common active learning scenario, a machine-learned model is initially trained on a minimal set of annotated seed data. The learned model is then used to analyse a large set of previously unseen examples, a set of maximally-informative examples is selected from this pool, and the selected set is annotated and added to the training data. The model is then retrained on the seed set plus the newly annotated examples, and the cycle repeats until some stopping point is reached. For an extensive survey of active learning methods and theory, see Settles (2009).

Sample selection strategies. Uncertainty sampling (Cohn et al., 1995) is one of the most widely-used AL selection method in natural language processing; it is also the one used in the current studies. Under this approach, examples are selected according to the uncertainty of the model regarding its

¹This chapter is based on and extends Baldridge and Palmer (2009) and Palmer et al. (submitted).

label predictions. Those examples about which the model is least confident are identified and then labeled by the annotator or oracle. Intuitively, if the model believes all possible analyses are more or less equally likely, it cannot confidently select one label over the others. The model’s low confidence level indicates that it has not had enough experience with that type of data to make an informed decision. Selecting high-uncertainty examples for annotation thus is intended to maximize the amount of new information provided for learning during each cycle.

In a related work, Settles and Craven (2008) discuss and evaluate active learning strategies for sequence prediction, introducing an information-density method designed to minimizing querying of outliers.

Other selection strategies make use of multiple learners. For example, query by committee (Seung et al., 1992) uses disagreement between multiple models as a diagnostic for informativeness of examples. Baldrige and Osborne (2008), on the other hand, use logarithmic opinion pools to select parses for manual disambiguation, using both standard uncertainty sampling and query by committee approaches.

Cost-sensitive selection. A recent development in active learning is cost-sensitive selection that is guided not only by the learner but also by the expected cost of labeling an example based on its likely complexity and/or the reliability of the annotator. Settles et al. (2008) provide empirical validation for cost-related intuitions; for example, that cost of annotation is static nei-

ther per example nor per annotator. Also, they show that taking annotation cost into account can improve active learning effectiveness, but that learning to predict annotation cost is not yet well-understood. A cost-sensitive Return on Investment heuristic is developed in Haertel et al. (2008a) and tested in a simulated POS-tagging context.

Measuring annotation cost. The usual case in active learning research is to simulate annotation using gold standard labels from the corpus. When an example is selected for annotation, it is added to the training set with the gold standard labeling.

Active learning studies with simulated annotation generally use a unit cost assumption that each word, sentence, constituent, document, or other selection unit takes the same amount of effort to annotate. This is often the only option since corpora typically do not retain and/or provide details of annotation time. In reality, examples can vary widely in the amount of effort required to label them. Assuming uniform annotation cost per example results in an inaccurate depiction of total annotation cost, skewing the presentation of annotation cost against accuracy gain. In active learning, assuming uniform cost is likely to exaggerate the annotation cost reductions achieved: the informative examples it seeks to find are typically harder to annotate (Hachey et al., 2005).

Baldrige and Osborne (2008) correlate a unit cost in terms of *discriminants* (decisions made by annotators about valid parses) to annotation

time. This is a better approximation than unit costs where such a relationship cannot be established.

Ideally, we would measure cost in terms of money spent since money paid to annotators will dominate annotation costs. This monetary expense in turn is usually dominated by annotator time, assuming annotators are being paid according to some temporal unit, so the time it takes to annotate each example is a good measure of actual cost.

In the live annotation experiments, we measure the exact time taken to annotate each example by each annotator and use this as the cost metric, inspired by Ngai and Yarowsky (2000). In the simulation studies, as we are unable to measure time, we measure cost by sentence/clause and word/morpheme.

Dynamic annotation cost. All of the cost measurement methods described above are based on *static* measurements of annotation time, and clearly the time taken to annotate an example is not a function of the example alone. Annotation time is actually *dynamic* in that it is dependent on how many and what kinds of examples have already been annotated. An “informative” example is likely to take longer to annotate if selected early than it would after the annotator has seen many other examples.

Thus, it is important to measure annotation time *embedded in the context* of a particular annotation experiment with the sample selection/labeling strategies of interest.

Active learning with real annotators. Though most active learning research to date simulates annotation, that trend seems to be changing. Ringger et al. (2007) evaluate both uncertainty sampling and query by committee for part-of-speech tagging and report that active learning helps in all cases. Various unit cost measurements are reported. Recent work by Settles et al. (2008) and Tomanek and Hahn (2009) measures actual annotation time for active learning for various NLP tasks. A slightly different take on the question is taken by Leidner (2007), who compares the time cost of five different approaches—including active learning—to constructing a set of named entity taggers.

Modeling the annotator. Another aspect of simulated studies is that they assume a single and infallible annotator. In real life, this is far from being the case. Donmez and Carbonell (2008) have begun to explore the question of how to model the annotator is sample selection, asking whether we get better results from active learning if examples are selected according to the characteristics of the annotator. Finally, work by Arora et al. (2009) studies the reliability of estimating annotation cost when multiple annotators are used in an active learning environment.

5.2 Organization of corpus for experimentation

The Uspanteko corpus described in Chapter 4 is used in the simulation experiments as well as in the live annotation experiments (Chapter 6). The

Section	Morphemes	Clauses	W/C	Texts
TRAIN	38802	8099	4.79	030,035,036,037,049,050,052,053,054,055 056,057,059,063,066,067,068,071,072,076,077
DEV	16792	3847	4.36	020,022,023,025,029
TEST	18704	3785	4.94	001,002,004,008a,014,016
TRANSL	7361			005,033
RAW	210157			003,006,007,009,010,011,012,013,017,018 019,021,024,026,027,031,032,034,041,047 048,060,061,062,064,069,070,073,074,075 080,081,110

Table 5.1: Corpus divisions.

corpus was divided into three subsets: 21 texts for training (**train**) material, 5 for development testing (**dev**), and 6 texts for post-development evaluation (**test**). Details of the corpus divisions appear in Table 5.1. It should be noted that these experiments use only those texts with full interlinear glossing. Two texts in the corpus have translations but no glossing (**trans**), and the remainder are transcribed but not translated (**raw**).

The corpus was split so as to maintain the original order of the texts in the corpus, indexed by the text ID. The specific division points were selected to maintain as well as possible the genre distribution and average clause length of the full corpus. The three subsets of the corpus are balanced with respect to average clause length, falling between averages of 4.36 and 4.94 morphemes per clause.

The subsets are *not* well-balanced with respect to genre. This is an artifact of maintaining the order of the original corpus, as the latter part of

Genre	All	Train	Dev	Test
STORY	48.70%	68.74%	23.95%	29.36%
PERSONAL EXPERIENCE	27.35%	8.24%	34.75%	60.37%
ORAL HISTORY	16.42%	16.74%	22.52%	10.27%
ADVICE	4.24%	0.00%	18.78%	0.00%
RECIPE	3.28%	6.29%	0.00%	0.00%

Table 5.2: Genre balance in corpus divisions

the corpus is predominantly of the story genre.² Table 5.2 presents the genre balance within the entire corpus and each subset; the figures shown are the percentage of morphemes in the data set from texts of the given genre. In the training set, the story genre is strongly overrepresented, and the personal experience genre is significantly underrepresented. In contrast, the story genre is underrepresented in both development and test sets, and the test set is heavily biased toward the personal experience genre.

For the simulation experiments, we also consider POS-tagging for Danish, Dutch, English, and Swedish; the English is from sections 00-05 (as training set) and 19-21 (as development set) of the Penn Treebank (Marcus et al., 1993), and the other languages are from the CoNLL-X dependency parsing shared task (Buchholz and Marsi, 2006).³ We split the original training data into training and development sets. Table 5.3 shows the number of words and sentences in each split of each dataset, as well as the number of possible labels and the average sentence length. The Uspanteko data is counted in morphemes

²See Section 4.1 for description of the four genres.

³The subset of the Penn Treebank was chosen to be of comparable size to the CoNLL datasets.

Language	#words tr	#words dev	# tags	#sents tr	#sents dev	avg sent	avg tr.sent	avg dev.sent
Danish	62825	31561	10	3570	1618	18.18	17.60	19.50
Dutch	129586	65483	13	9365	3982	14.61	13.84	16.44
English	167593	131768	45	6945	5527	24.00	24.13	23.84
Swedish	127684	63783	41	7326	3714	17.34	17.43	17.17
Uspanteko	43473	19906	69	7423	3288	5.92	5.86	6.05

Table 5.3: Corpora: number of words and sentences, number of possible tags, and average sentence length.

rather than words; also, the Uspanteko texts are divided at the clause rather than sentence level. This gives the corpus a much lower average clause length than the other languages (Table 5.3).

The OKMA Uspanteko data are the result of an annotation effort focused on producing fully interlinearized and glossed texts. Though they are machine-readable, the original annotations are designed primarily to be useful for human users. Furthermore, the original gloss labels were determined by a number of different annotators with different levels of linguistic experience and training. As a result, even after extensive clean-up, the Uspanteko dataset exhibits more noise and inconsistency than the usual datasets used for natural language processing.

Finally, while the Uspanteko datasets are very small by the standards of computational linguistics, the size is realistic for computational work on endangered languages.

5.3 Model and methods

In this section we describe the models and methods used for all active learning experiments, both with simulated annotation and with live annotation. The experiments use simple, strong, and standard approaches to both classification and sample selection, for two reasons. First, the focus of the research is on the interaction between the annotator and different levels of machine involvement. Second, the methods employed might feasibly be adopted by a field linguist with minimal support from a computational linguist.

Classification model. We use a standard maximum entropy classifier for tagging Danish, Dutch, English, and Swedish words with POS-tags and tagging Uspanteko morphemes with Gloss/POS tags. The label for a word/morpheme is predicted based on the word/morpheme itself plus a window of two units before and after. Standard part-of-speech tagging features (Ratnaparkhi, 1998; Curran and Clark, 2003) are extracted from the morpheme to help with predicting labels for previously unseen morphemes. This is a strong but standard model; better, more complex models could be used, but the gains are likely to be small. Thus, we opted for simplicity in our model so as to focus more on the interaction between the annotator and different levels of machine involvement.

The accuracy of the tagger on the datasets when trained on all available training material is given in Table 5.4, along with accuracy of a unigram model (learned from the training set and constrained by a tag dictionary for known

	Unigram	Model
Danish	91.62%	95.58%
Dutch	90.92%	93.57%
English	87.87%	93.25%
Swedish	84.91%	87.74%
Uspanteko	77.84%	79.39%

Table 5.4: Training on all data: unigram probability and model performance

words).⁴ Accuracy is measured as absolute performance.

Sample selection. The sample selection method being evaluated is uncertainty selection. Uncertainty selection Cohn et al. (1995) identifies examples which the model is least confident about. We measure uncertainty as the entropy of the label distribution predicted by the maximum entropy model for each example. Uncertainty for a clause is calculated as the average entropy per morpheme; clauses with the highest average entropy are selected for labeling.

We compare uncertainty (**unc**) selection against two baseline methods: sequential (**seq**) and random (**rand**). For reasons of coherence and the importance of context, the default annotation procedure in language documentation is sequential selection. So it is important for us to compare our learner-guided selection to business-as-usual, even though random selection generally works better. However, sequential selection is generally sub-optimal, particularly for corpora with contiguous sub-domains (e.g. texts from different genres),

⁴Note that the English performance is based on using only sections 00-05 of the Penn Treebank, so accuracy is lower than with the usual practice of training on 00-18.

because it requires annotation of many similar examples in order to get to examples that, due to their novelty, are likely to help a learned model generalize better. Random selection requires no machine learning but typically works much better than sequential selection. Random avoids the sub-domain trap by sampling freely from the entire corpus, and it provides a strong baseline against which to compare learner-guided selection, such as uncertainty sampling. In addition, random selection is at times competitive with more involved active learning methods and is often better early on in the annotation process (Baldrige and Osborne, 2008).

5.4 Simulation experiments

Simulation experiments verify that our tagger and dataset behave as expected in standard active learning conditions. We run simulations for morpheme labeling on the Uspanteko data set, and on POS-tagging for Danish, Dutch, English, and Swedish. Here, we vary only the selection method: **sequential**, **random**, or **uncertainty**.

5.4.1 Parameters and evaluation

For each language, we randomly select a seed set of 10 labeled sentences. The number of examples selected to be labeled in each round begins at 10 and doubles after every 20 rounds. For **rand** and **unc**, each batch of examples is selected from a pool (size of 1000) that is itself randomly selected from the entire set of remaining unlabeled examples. Using a pool is important for

the **unc** conditions, in which the learned model has to label every unlabeled example for selection can occur. When examples are selected from a pool rather than from the full unlabeled set, the time required for labeling examples is greatly reduced. **rand** and **unc** experiments for each language are replicated 5 times; performance of a particular selection method is evaluated (as described below) by computing splines and regressions over all runs for each condition.

Learning curve comparison. We are interested in comparative evaluation of many different experimental settings, across which we vary selection methods and, in the live annotation experiments, use of label suggestions and annotators. To achieve this, it is useful to have a summary value for comparing the results from two individual experiments. One such measure is the percentage error reduction (PER), measured over a discrete set of points on the first 20% of the points on the learning curve (Melville et al., 2004).⁵

We use a new related measure, which we call the *overall* percentage error reduction (**OPER**), that uses the *entire* area under the curves given by fitted nonlinear regression models rather than averaging over a subset of data points. Specifically, we fit a modified Michaelis-Menton model:

$$f(cost, (K, V_m, A)) = \frac{V_m(A + cost)}{K + cost}$$

⁵This is justified in standard conditions, sampling from a finite corpus: active learning runs out of interesting examples after considering a fraction of the data, so the curve is *artificially* pulled down by the remaining, boring examples.

The (original) parameters V_m and K respectively correspond to the horizontal asymptote and the cost where accuracy is halfway between 0 and V_m . The additional parameter A allows for a better fit to our data by allowing for less sharp elbows and letting *cost* be zero. Model parameters were determined with `nls` in R (Ritz and Streibig, 2008).

With the fitted regression models, it is straightforward to calculate the area under the curve between a start cost c_i and end cost c_j by taking the integral from c_i to c_j . The overall accuracy for the experiment is given by dividing that area by $100 \times (c_j - c_i)$. Call this the overall curve accuracy (OCA). Then, for experiment A compared to experiment B , $\text{OPER}(A,B) = \frac{OCA_A - OCA_B}{100 - OCA_B}$. For the simulation experiments we calculate OPER for only the first 20% of cost units. For the annotation experiments, we calculate it for the minimum amount of time spent on any of the experiments (which ended up using less than 10% of all available morphemes).

5.4.2 Results

Figure 5.1 gives learning curves for the Uspanteko simulations, with cost measured in terms of (a) clauses and (b) morphemes. Both graphs show the usual behavior found in active learning experiments. `Srand` and `unc` both rise more quickly than `seq`, and `unc` is well above `rand`. The relationship between the methods is the same regardless of the cost metric, but the relative differences in cost-savings are not, which we see when we look at OPER values.

The dashed vertical lines in the two graphs correspond to the 20% mark

	$\frac{\mathbf{rand}}{\mathbf{seq}}$	$\frac{\mathbf{unc}}{\mathbf{seq}}$	$\frac{\mathbf{unc}}{\mathbf{rand}}$
Uspanteko-Clauses	5.86	13.27	7.86
Uspanteko-Morphs	7.47	11.68	4.55

Table 5.5: OPER values for Uspanteko simulations, comparing **clause** and **morpheme** cost. $\frac{A}{B}$ indicates we compute $\text{OPER}(A,B)$.

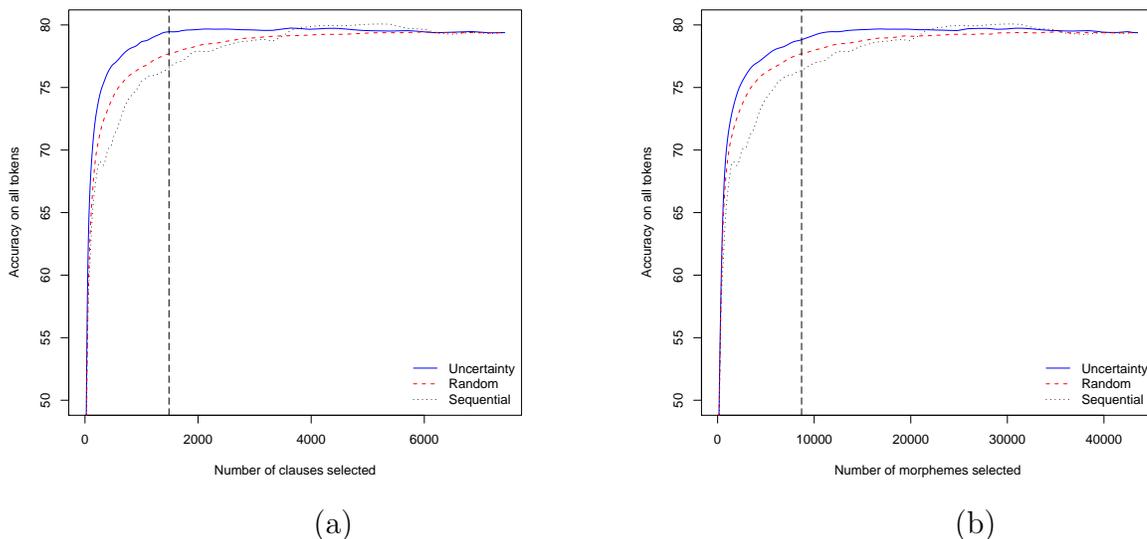


Figure 5.1: Learning curves for Uspanteko simulations; (a) clause cost and (b) morpheme cost.

used to calculate OPER values, which are given in Table 5.5. Specifically the dashed lines indicate: (a) 1485 clauses, and (b) 8695 morphemes. Most importantly, note the much larger OPER for **unc** over **rand** with clause cost (7.86 vs 4.55). Also note that $\text{OPER}(\mathbf{rand},\mathbf{seq})$ is *lower* with clause cost—this indicates that the beginning portions of the corpus contain longer sentences with more morphemes, an accident which overstates how well **seq** would likely

	rand seq	unc seq	unc rand
Danish	4.58	6.95	2.48
Dutch	21.95	23.68	2.20
English	6.55	8.00	1.56
Swedish	9.56	9.29	-0.30
Uspanteko	7.47	11.68	4.55

Table 5.6: OPER values for **morpheme** cost for simulations. $\frac{A}{B}$ indicates we compute $\text{OPER}(A, B)$.

work in general.

Since **rand** is unbiased with respect to picking longer sentences, the large increase of $\text{OPER}(\mathbf{unc}, \mathbf{rand})$ from 4.55 to 7.86 is a clear indication of the well-known—but not always attended to—tendency of uncertainty sampling to select longer sentences. Consequently, one should at least use sub-sentence cost in order not to overstate the gains from active learning. The live annotation experiments in Chapter 6 take this word of caution one step further: even sub-sentence cost (morpheme cost, in our setting) can overestimate gains since the clauses selected are actually harder to annotate and thus take more time.

Table 5.6 gives overall percentage error reductions (OPER) between different selection methods based on word/morpheme cost, for each language. For all languages, **rand** and **unc** are better than **seq**. Only in the case of Swedish is there no benefit from **unc** over **rand**. For Dutch, both **rand** and **unc** show large gains over **seq**. This reflects the heterogeneity of the underlying Alpino corpus,⁶ as sequential selection is likely to assemble training

⁶<http://www.let.rug.nl/vannoord/trees/>

data from the types of texts found early in the corpus but then test on data from other types of texts. Most importantly, for Uspanteko, there are large reductions from **unc** to **rand** to **seq**, mirroring the clear trends in Figure 5.1b.

These simulations have an unrealistic “perfect” annotator, the corpus. Next, we discuss results with real annotators—who may be fallible or may (reasonably) beg to differ with the corpus analysis.

Chapter 6

Active learning with live annotation

In simulated studies such as those presented in Chapter 5, active learning has been shown for many tasks to reduce the amount of training data needed to reach a given level of performance. However, the amount of data labeled is not necessarily an accurate predictor of actual annotation cost. In this chapter we show that in a real annotation scenario, measuring cost by the **time** spent on annotation produces rather different results than are seen with unit-cost measurements. Although time is the main factor determining cost in most scenarios, and thus an appropriate measure of annotation effort, one could pay an annotator per unit labeled, as with Amazon Mechanical Turk¹ and similar systems.

The experiments presented in this chapter assess the potential of learner-guided example selection and machine label suggestion to enable more efficient production of interlinear glossed text (IGT, Chapter 3). We report on timed annotation experiments that vary three factors: annotator expertise, example selection methods, and suggestions from a machine classifier.²

¹<http://www.mturk.com>

²This chapter is based on and extends Palmer et al. (2009) and Baldrige and Palmer (2009).

The experiments represent an intermediate stage between simulated annotation and actual practical application of the techniques to never-before-labeled material. We use the same Uspanteko corpus used for the simulation studies, but examples selected for labeling are annotated by human annotators.

6.1 Annotators

In this study we compare annotations performed by two annotators. Both are linguists who specialize in language documentation and have extensive field experience. Both are fluent speakers of Spanish, the target translation and glossing language for the OKMA texts. One annotator had extensive prior experience with Uspanteko (the language expert, **LgExp**), and the other had none (the language novice, **LgNov**).

The **LgExp** is a doctoral student in language documentation who has done extensive linguistic and lexicographic work on Uspanteko. Her work includes a written grammar of the language (Can et al., 2007b) and contributions to the publication of an Uspanteko-Spanish dictionary (Vicente Méndez, 2007). Additionally, she is a native speaker of K'ichee', a Mayan language that is closely related to Uspanteko.

The **LgNov** is a doctoral student in language documentation whose work focuses on indigenous languages of Mesoamerica, particularly Chatino and Zapotec. At the start of the annotation studies, he had no previous experience with Uspanteko and only limited prior knowledge of the structure of Mayan languages. He had access to the Uspanteko-Spanish dictionary during

annotation, but not to the grammar.

6.1.1 Annotator expertise

These two annotators were chosen specifically for their different levels of expertise in the language. The time of a linguist with language-specific expertise is one of the most valuable resources for producing IGT, and our experiments touch on the question of how to most efficiently use that resource in the annotation process. But documentation projects often draw also (or sometimes instead) on the time of linguists *without* prior experience in the language. We compare the relative effectiveness of machine support for these two different types of annotators and find evidence that expertise influences which selection strategies are most effective.³

A factor related to expertise is that not all annotators cost the same. For example, the most knowledgeable and possibly most efficient annotator might well be the most costly or have the most limited time (which has the same effect, for language documentation). This sort of factor would ideally inform an active learning process, though we do not address it here.

³It should of course be noted that one annotator per type, as we have in these studies, is too small a sample to draw generalizable conclusions. Our results are suggestive but not conclusive. At the same time, the two-annotator scenario accurately reflects the resources available to many documentation projects.

6.1.2 Annotator training and learning curve

A similarity of our setup to a typical documentation project is the absence of a detailed annotation manual. Annotation in language documentation is itself a process of discovery. Analyses change as annotation proceeds, and annotation conventions necessarily change along with them.⁴

Even without strict guidelines, though, annotators need to have some sense of common conventions, and in particular our annotators needed to have some sense of the conventions of the original OKMA annotations. To this end, we use a new annotator training process.

Two seed sets of ten clauses each were selected to be used both for human annotation training and for initial training of the machine learners. In separate sessions, each annotator was given these morpheme-segmented clauses to label, one set of ten at a time. The labels were compared to the original OKMA labels, and results indicating correct and incorrect labels were shown. The annotator’s task was to relabel all incorrect labels, iterating the process until the two sets of labels matched completely. In cases where the annotator made 5–7 consecutive incorrect guesses, the correct label was provided.

The first ten clauses of the first text in the training data were used to seed model training for the sequential selection cases. The second set of ten were randomly selected from the entire corpus and used to seed model training for both random and uncertainty sampling.

⁴For further discussion, see Section 2.3.

In any annotation project, annotators go through an initial phase during which they become familiar with the data, the annotation guidelines and the annotation interface. During this phase, per-label annotation time is generally higher than it is later in the process, and mistakes and inconsistencies are more likely to occur. While annotation times for the **LgExp** line up with this typical case, for the **LgNov** the learning curve is much steeper; in addition to familiarization with guidelines and interface, he is in fact discovering the nature of the language as he goes.

As expected, **LgExp**'s learning curve reaches a plateau far more quickly than that for **LgNov**. Her learning process consisted primarily of remembering aspects of the earlier analysis of Uspanteko (i.e. the analysis reflected in the grammar), noting subsequent changes in analysis, and resolving some inconsistencies in her labeling choices. The **LgNov**, starting from zero, needed much more work to acquire proficiency with the language and task. This is reflected in his average annotation time per morpheme, shown in Figure 6.1.

6.2 Annotation infrastructure

Incorporating machine label suggestions into an annotation infrastructure requires careful attention to implementation, especially concerning the annotator's interaction with the system.

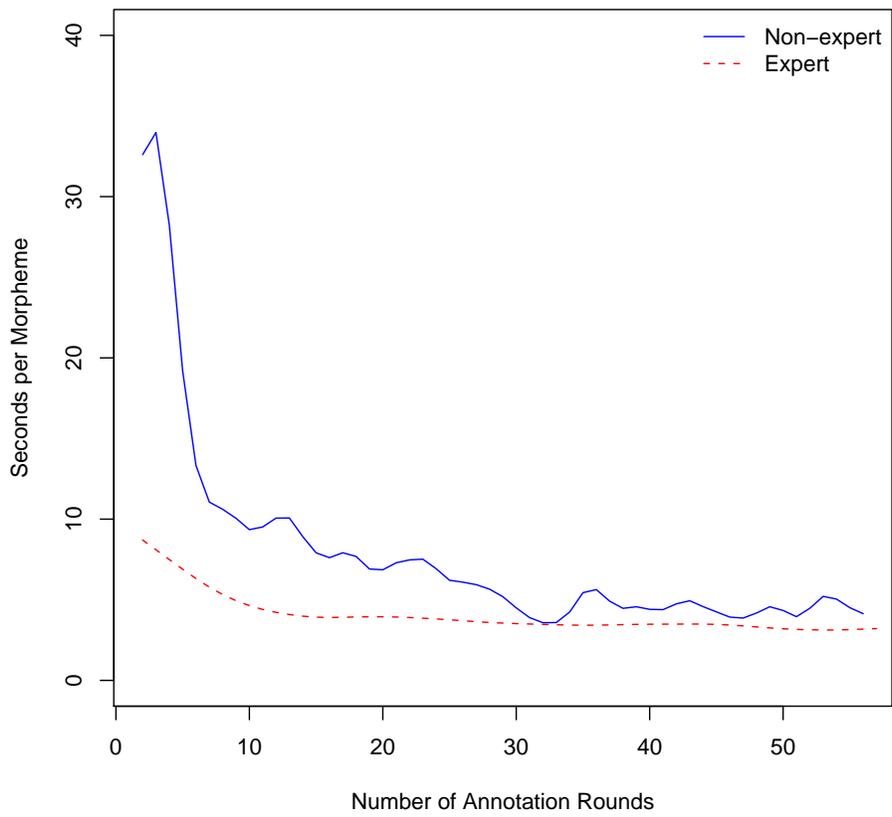


Figure 6.1: Average annotation time (in seconds per morpheme) over annotation rounds, averaged over all six conditions for each annotator.

6.2.1 Label suggestions

One way of investigating the effectiveness of machine support for the annotation process is through providing predictions of the model as suggestions to the annotator. This reduces the number of decisions to be made, as correctly-labeled items require no action from the annotator. Semi-automated labeling of this nature can dramatically reduce annotation costs (Baldrige and Osborne, 2004).

The idea of using label suggestions is quite straightforward: the model ranks the possible labels which it might assign to a morpheme, and the annotator uses that ranked list, rather than the full, uninformative list of all possible labels, to come to a determination more quickly. Ideally, the right label is ranked at the top of the list and is thus the first label provided, meaning the annotator just needs to spot-check the model output.

Our experiments consider two conditions for providing classifier labels: a **do-suggest (ds)** condition where the labels predicted by the machine learner are shown to the annotator, and a **no-suggest (ns)** condition where the annotator does not see the predictions. The **ds** cases show the annotator the most probable label according to the most-recently-learned model, as well as a ranked list of other highly-likely labels.⁵ In the **ns** cases, the annotator is shown a list of labels previously seen in the training data for the given morpheme; this list is ranked according to frequency of occurrence. Note that

⁵To appear on this list, a label must be at least half as probable as the best label.

this is a stronger no-suggest baseline than one which simply lists all labels in alphabetical order. Providing the list of previously-seen labels in the **ns** conditions is intended to mirror an annotator’s interaction with Shoebox/Toolbox, making for a better comparison to typical language documentation methods.

6.2.2 Annotation tool

Evaluating the effectiveness of machine support in different experimental conditions (Section 6.3.1) requires integrating automated analysis into the manual annotation process. The integration requires careful coordination of three components: 1) presenting examples to the annotator and storing the annotations, 2) training and evaluating tagging models using data labeled by the annotator, and 3) selecting new examples for annotation. Since no existing annotation tool directly supports such integration, we developed a new tool, the OpenNLP IGT Editor⁶, to manage the three processes. The annotation component of the tool, and in particular the user interface, is built on the Interlinear Text Editor (Lowe et al., 2004).⁷

An example of annotating a clause with the IGT editor is given in Figure 6.2. The editor window displays the static tiers of the IGT annotation for the clause; these are the TEXT, TRANS, and MORPH lines. The first two appear in the upper left window of the editor. The individual morphemes are presented for labeling in the window below, grouped by word, with a separate

⁶<http://igt.sourceforge.net/>

⁷http://michel.jacobson.free.fr/ITE/index_en.html

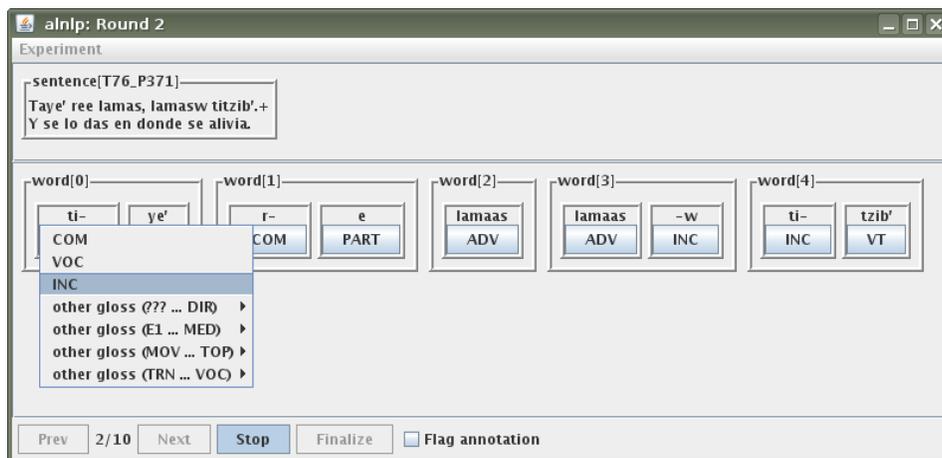


Figure 6.2: The OpenNLP IGT Editor interface.

gloss field for each morpheme.

This particular example shows the state of the editor as an annotator labels the first morpheme of a clause in one of the **ds** conditions. The clause initially displays with the gloss fields populated by the most-likely label for each morpheme, as determined by the learned classification model. In this case, the annotator has not immediately accepted the machine’s label suggestion and instead seeks to choose a different label. The label choices appear in a drop down menu for the gloss field. The first three items on the menu—**COM**, **VOC**, and **INC**—are label suggestions from the machine, ranked by decreasing likelihood. The rest of the label set is accessible through the alphabetically-organized menus appearing below the label suggestions. Every label in the pre-determined label set is available for every morpheme, but a few have been highlighted by the machine as more likely choices. One advantage of using a

fixed label set presented in drop down menus is that it prevents label inconsistencies by not allowing free input.

The annotation tool also measures and logs the time taken to annotate each individual clause. The menu bar at the bottom of the editor window both tracks progress through the batch of clauses (shown by the **2/10** counter) and gives the annotator the ability to stop timing in order to take breaks. When the annotator hits the **Stop** button, though, the screen greys out and the clause is no longer visible. The editor also allows free movement between clauses in the batch, but no revision is possible once the annotations for the batch have been finalized. The final point to note is the **Flag annotation** checkbox at the bottom center of the window. In an ideal tool, the annotator would be able to change segmentations as well as making gloss label decisions, but the OpenNLP IGT Editor does not offer that flexibility. As a compromise, the checkbox allows the annotator to flag clauses with problematic segmentations and/or analyses for later inspection. The editor processes IGT-XML, and the set of flagged clauses is easily retrievable from the XML files.

An additional requirement was that the editor interface had to be intuitive and easy-to-use. Anticipating and handling the users' needs, particularly those of the novice user, added significantly to the development time. Yet still some human-computer interaction issues turned out to hurt performance (in terms of accuracy per the amount of time spent) for both the learned models and the human annotators. This is discussed in greater detail in Chapter 7.

6.3 Annotation experiments

In this section we present experimental conditions, evaluation metrics, and results for extensive active learning experiments with live annotation.

6.3.1 Experimental conditions

The individual experiments vary from one another in three aspects: sample selection method, machine label suggestions, and expertise of annotator. The selection methods used in these experiments are those described in Section 5.3. Machine label suggestions and annotator expertise are discussed in Section 6.1 and Section 6.2.1, respectively. Having two annotators (**LgExp**, **LgNov**), three selection methods (**seq**, **rand**, **unc**), and two machine labeling settings (**ns**, **ds**) results in a total of 12 different experiments.

Annotators improve as they see more examples. To minimize the impact of this learning process, annotation is done in rounds. Each round consists of sixty clauses—six batches of ten each for the six experimental cases. The annotator is free to break between batches. Following annotation, the newly-labeled clauses are added to the training data, and a new model is trained and evaluated. Both annotators completed fifty-six rounds of annotation.

6.3.2 Results

Our focus is on overall accuracy with time cost. First, though, we briefly summarize results when cost is measured by the number of clauses labeled. In the current experiments, clause cost shows the same patterns noted in the

simulation experiments in overestimating the gains from active selection. We discuss morpheme cost as well, primarily to show that (a) it can misstate relative effectiveness when comparing annotators, and (b) like clause cost, it overstates gains from active selection.

To start, we consider the results of the timed annotation studies in broad strokes, highlighting the key results. We then look at the effectiveness of different strategies for a single annotator. Finally, we present learning curve comparisons varying each of the three variables.

The big picture Figure 6.3 shows curves for four experiments: **seq-ns** for both annotators⁸ and the most effective overall condition for each annotator (**rand-ds** for the **LgNov**, **unc-ns** for the **LgExp**).

Figure 6.3a uses morpheme cost evaluation; on that metric, both annotators appear to be about equally effective with **seq-ns** and much more effective with machine involvement (**unc** or **ds**) than without. Additionally, the **LgNov**'s **rand-ds** appears to beat the **LgExp**'s **unc-ns**. However, the time cost evaluation in Figure 6.3b tells a dramatically different story. For a given annotator, morpheme cost and time cost agree as to the relative effectiveness of various strategies. The impact of switching to time cost appears when we compare different annotators.⁹ Each annotator's machine-involved

⁸Recall that sequential annotation is the default mode for producing IGT, so this strategy is of particular interest.

⁹It is also clear to see that, unsurprisingly, the **LgExp** spent much less time to complete the 56 rounds than the **LgNov**. In general, the **LgExp** annotator was much quicker,

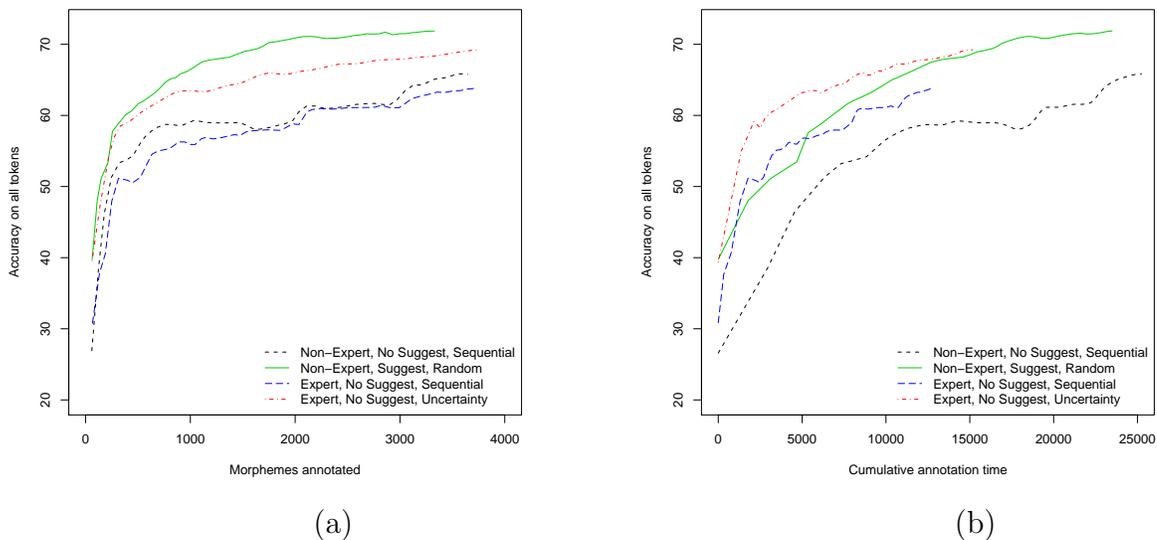


Figure 6.3: A sample of the learning curves with (a) morpheme cost and (b) time cost.

experiment is much better than their **seq-ns**, but now the **LgExp**'s best is clearly better than the **LgNov**'s. We see this as clear evidence for the need for cost-sensitive learning in which the expected cost of annotation plays a role in the sample selection process (Haertel et al., 2008b; Settles et al., 2008).¹⁰

The **LgNov** with **rand-ds** caught up to and surpassed the unaided **LgExp** in about six hours total annotation time, and he caught up to her highest-performing curve (**unc-ns**) after 35 hours. This is encouraging since often language documentation projects have participants with a wide range of expertise levels, and these results suggest that assistance from machine

particularly in early rounds, averaging 4.1 seconds per morpheme annotated against the **LgNov**'s 8.0 second average.

¹⁰Also note that the current experiments do not use cost-sensitive learning methods.

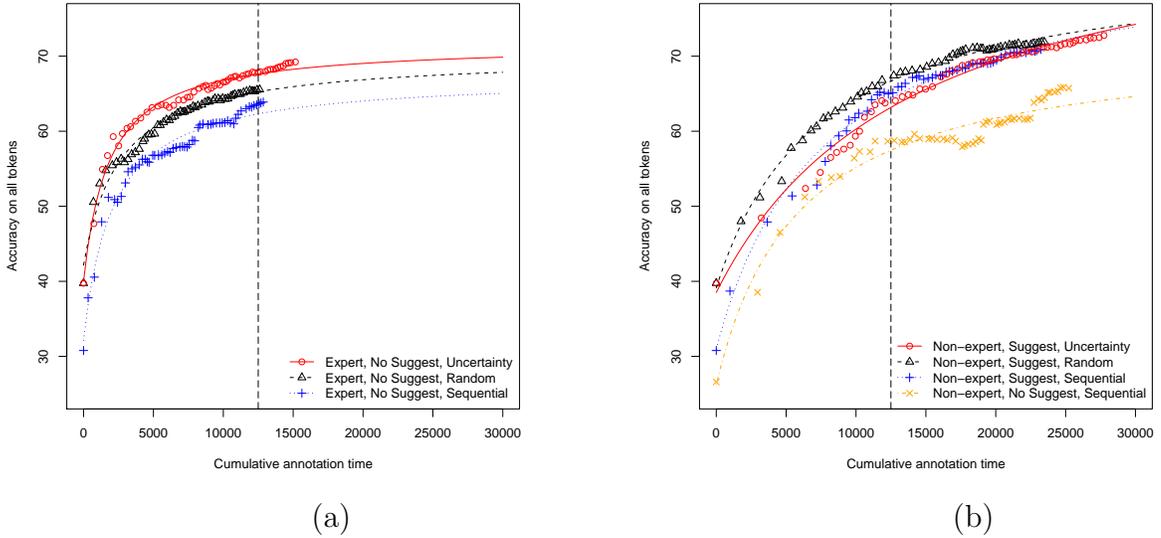


Figure 6.4: Sample measurements and fitted nonlinear regression curves for (a) the **LgExp** and (b) the **LgNov**. Note that the scale is consistent for comparability.

learning, if done properly, may increase the effectiveness of participants with less language-specific expertise. We are also encouraged, with respect to the effectiveness of active learning, that the **LgExp**'s best performance is obtained with uncertainty-based, learner-guided selection.

Within-annotator comparisons Figure 6.4¹¹ shows both actual measurements and the fitted nonlinear regression curves used to compute OPER. Figure 6.4a, the **LgExp** without suggestions, exhibits typical active learning behavior similar to that seen in the simulation experiments. Figure 6.4b, the

¹¹The dashed vertical lines indicate 12,500 seconds (about 35 hours), which is the upper limit used in computing all OPER values presented in this chapter.

	LgExp over LgNov
seq-ns	15.99
rand-ns	11.52
unc-ns	10.76
seq-ds	12.34
rand-ds	7.67
unc-ds	19.13

Table 6.1: OPER for language expert over language novice.

LgNov *with* suggestions, shows that in the **ds** conditions the **LgNov** was less effective with **unc**. This is not unexpected: uncertainty selects harder examples that will either take longer to annotate or are easier to get wrong, especially if the annotator trusts the classifier and *especially* on examples the classifier is uncertain about. Nonetheless, in all **ds** cases, the **LgNov** performs better than with **seq-ns**.

Learning curve comparisons. The next three tables provide OPER¹² values from time 0 to 12,500 seconds (about 35 hours), the minimum amount of annotation time logged in any one of the twelve experiments.¹³ Table 6.1 gives OPER for the **LgExp** versus the novice given the same selection and suggestion conditions. Table 6.2 gives OPER for the **LgExp** versus herself for different conditions, and Table 6.3 shows the same for the language novice. For example: (Table 6.1) the **LgExp** obtained an 11.52 OPER versus the novice

¹²Details of the overall percent error reduction (OPER) metric can be found in Section 5.4.1.

¹³Stopping at 12,500 seconds ensures a fair comparison, for example, between the **LgExp** and the **LgNov** because it requires no extrapolation of the **LgExp**'s performance.

LgExp \ LgExp	seq-ns	rand-ns	unc-ns	seq-ds	rand-ds	unc-ds
seq-ns	—					
rand-ns	8.85	—				
unc-ns	14.17	5.83	—			
seq-ds	6.34	-2.76	-9.12	—		
rand-ds	10.52	1.83	-4.25	4.46	—	
unc-ds	14.50	6.20	0.39	8.72	4.45	—

Table 6.2: OPER comparisons using time cost measurement, language expert.

LgNov \ LgNov	seq-ns	rand-ns	unc-ns	seq-ds	rand-ds	unc-ds
seq-ns	—					
rand-ns	13.46	—				
unc-ns	19.20	6.63	—			
seq-ds	10.24	-3.72	-11.09	—		
rand-ds	18.59	5.93	-0.76	9.30	—	
unc-ds	11.19	-2.62	-9.91	1.06	-9.09	—

Table 6.3: OPER comparisons using time cost measurement, language novice.

when both used **rand-ns**; (Table 6.2) the **LgExp** obtained a 10.52 OPER by using **rand-ds** rather than **seq-ns**; and (Table 6.3) the novice obtained a 5.93 OPER over **rand-ns** by using **rand-ds**.

A number of patterns emerge. Unsurprisingly, the values Table 6.1 show that the **LgExp** is more effective than the novice in all conditions. Also, every other condition is more effective than **seq-ns** for both annotators (first column in both Table 6.2 and Table 6.3). **unc-ns** and **rand-ds** are particularly effective for the **LgNov**, giving OPERs of 19.20 and 18.59 over **seq-ns**, respectively. These reductions, bigger than the **LgExp**'s reductions of 14.17 and 10.52 for the same conditions, considerably reduce the large gap in **seq-ns** effectiveness between the two annotators (see Figure 6.3b).

The **LgExp** actually gains very little from **ds** for both **rand** and **unc**:

adding suggestions gave OPERs of just 1.83 and .39, respectively. In contrast, the **LgNov** obtains an improvement of 5.93 OPER when suggestions are used with **rand**, but performs *worse* when used with **unc** (-9.91 OPER). Even more striking: the **LgNov**'s **unc-ds** is worse than **rand-ns** (-2.62 OPER), a completely model-free setting. These variations demonstrate the importance of modeling annotator fallibility and sensitivity to cost, as well as characteristics of the annotation task itself, if learner-guided selection and suggestion are to be used (Donmez and Carbonell, 2008; Arora et al., 2009).

6.3.3 Discussion

One of the key findings from these experiments—one which we fully expected—is that it is imperative to measure cost in terms of time rather than using a unit cost. This is crucial since unit cost is the standard practice in active learning studies (which are almost entirely simulation studies). Measuring cost in terms of morphemes indicated that the **LgNov** was the most effective annotator, but this result reversed when the time used to annotate was taken into account: with time cost, the **LgExp** produced datasets that trained more accurate classifiers much more quickly.

Our experiments do not employ cost-sensitive selection; in fact, our results—from a live (non-simulated) active learning experiment of moderate scale—can be taken as providing empirical support for the need to consider cost-sensitive selection in order to ensure better cost reductions.

The second, more surprising, finding is that uncertainty selection worked

well with **LgExp**, but it performed worse than random selection with **LgNov**. Returning to Figure 6.4b, we see that random (indicated by triangles) performs *much* better than uncertainty (circles) until rather late in the annotation process, and we do not see the early gain from learner-guided selection that is seen in most other work on active learning. As previously mentioned, the examples selected by uncertainty sampling tend to be harder to annotate. They are particularly challenging for the **LgNov**, who lacks the language-specific knowledge needed to handle difficult cases. The labels he provides for these cases are more likely to be incorrect than for any of the other cases (see Table 7.1), so the associated model is given many bad labels as part of its training data. This results in a less accurate model. The advantage of random sampling for the **LgNov** is that more of the selected examples will be easier to annotate and/or similar to things he has already seen. His labels are thus more accurate, resulting in better training data and a better model.

This indicates that language (or domain) expertise matters in using active learning. In particular, it indicates that we must develop methods that model not only how useful any given example is likely to be (e.g., using uncertainty), but also how well and how quickly a given annotator is likely to annotate it. There has been very little work on annotator-aware selection strategies in active learning research so far, yet it is clearly essential if active learning is to be an effective technique in real-life annotation projects.

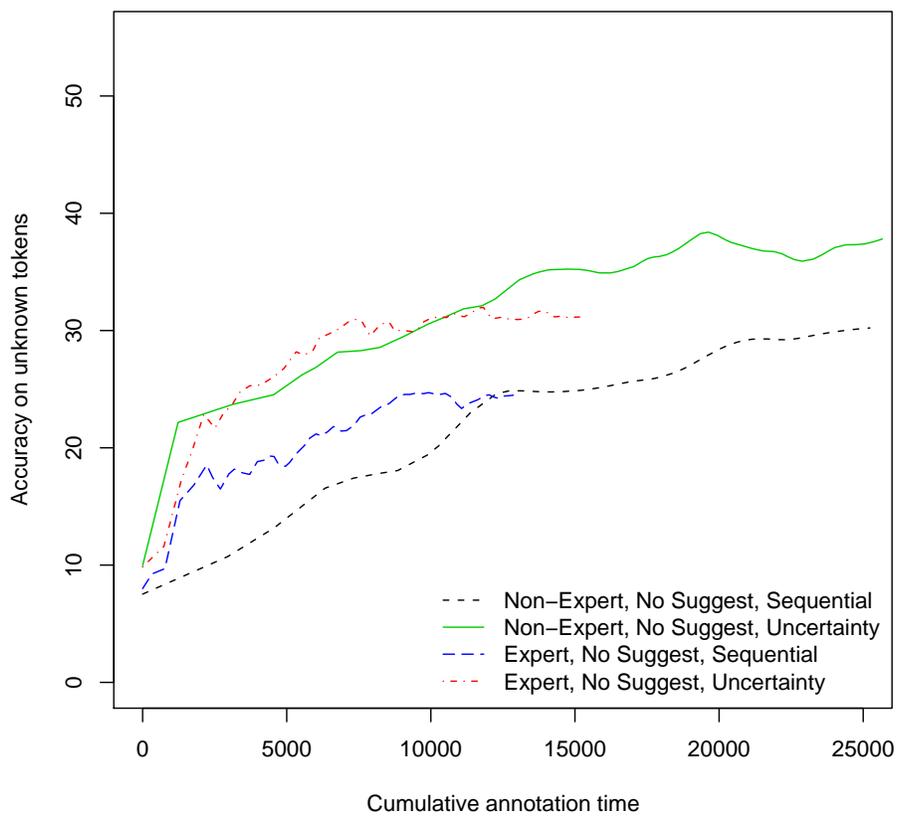


Figure 6.5: Accuracy on previously-unseen morphemes for both annotators, **seq** vs. **unc**.

This discussion of selection strategy effectiveness pertains to the accuracy of the learned model in labeling all words in the corpus, but this is just one way to measure the adequacy of the models and of the entire labeled set. For example, improved performance on uncommon constructions might be more important than overall high accuracy on the common cases. Figure 6.5 shows that prediction of labels for unseen morphemes gets a particularly large boost from active learning. This is highly relevant for language documentation: a major goal is to analyse the long tail of words/constructions in the language that may not be common but are linguistically interesting.

The third major finding is that label suggestions provided by the machine were useful for **LgNov** but not for **LgExp**. **LgNov** found the suggestions useful for limiting the likely analyses for a given morpheme, whereas **LgExp** initially found them to be a distraction and only paid attention to them later on in the annotation when the machine's predictions had become more accurate. However, these results were somewhat confounded by an unforeseen human-computer interaction issue. The way that label suggestion was implemented unexpectedly made it harder at times for the annotators to locate the label they wanted to select. The primary observation on label suggestions, then, is that it is probably most important to consider the interface design when hoping to allow machine suggestions to speed up annotation. The other, unsurprising, observation is that machine label suggestions should only be provided after the machine is sufficiently accurate. This suggests that there should be studies on measuring when a machine classifier is sufficiently

accurate to begin suggesting labels (this is not a trivial thing to do, since early in a project there would be no evaluation set available).

In summary, the standard strategy of sequential annotation with no input from a machine learner is outperformed by some configurations of learner-guided example selection and machine label suggestions. However, annotators with different levels of expertise may find different strategies to be more or less effective when it comes to quickly and efficiently producing a fully-labeled corpus of a given accuracy. The impact of differences between annotators indicates that in order to reliably obtain cost reductions with active learning techniques, annotators' fallibility, reliability, and sensitivity to cost must be modeled (Donmez and Carbonell, 2008). The results also bring into focus the general uncertainty regarding how well active learning works in practical applications (Tomanek and Olsson, 2009). This is particularly important in the language documentation context, where software support for documenting languages has to be robust, flexible, easy to learn, and straightforward to use.

Chapter 7

Complexities of active learning with live annotation

Applying active learning in a real annotation scenario, as we have done here, introduces human factors which play no role in simulated active learning studies. This chapter discusses three such factors. First, human annotators are fallible and may make mistakes in labeling. Second, human annotators do not always agree about the best label for a given item. Working with multiple annotators on a project may result in assembling a set of labeled data with internal contradictions. And third, the interactions of the human with both the annotation tool and the machine learner, as well as the effect of those interactions on the learning process, are not yet well-understood.¹

7.1 Annotator accuracy and consistency

One factor which must be considered when annotation is done by human annotators rather than being simulated is the accuracy of the humans' labels. Table 7.1 shows the overall accuracy of the two annotators' labels for

¹This chapter includes material previously published in Baldrige and Palmer (2009) and contained in Palmer et al. (submitted). Much of the material in Section 7.3 is the outcome of discussions with the annotators, Eric Campbell and Telma Can.

	LgExp	LgNov
seq-ns	73.17%	75.09%
rand-ns	69.90%	74.37%
unc-ns	61.23%	60.04%
seq-ds	67.48%	73.13%
rand-ds	68.34%	73.03%
unc-ds	59.79%	60.27%

Table 7.1: Overall accuracy of annotators’ labels, measured against OKMA annotations.

each condition after 56 rounds. Accuracy is measured in terms of percent agreement with the original OKMA annotations. The first result to note is that accuracy for both annotators suffers in both **unc-ns** and **unc-ds**. This is not unexpected; **unc** sampling picks examples that are more difficult to annotate (Hachey et al., 2005).

More surprising is the fact that the **LgNov**’s accuracies are generally higher than the **LgExp**’s; this is another result which highlights the differences and challenges that arise when we bring active learning into non-simulated annotation contexts. We attribute this result to two different factors. The first is the speed of annotation; the **LgNov** spent nearly twice as much time labeling the same number of examples, so each one was done with more care. The more interesting factor, though, again has to do with prior experience with Uspanteko. The typical assumption is that the original or reference corpus is the gold standard—a true, fixed target against which annotator or machine-predicted labels should be measured. In language documentation, though, the analysis of the language is continually evolving, and analysis and annotation

each inform the other. In fact, **LgExp** recognized (in the morphological segmentation) several linguistic phenomena for which the analysis has changed since the close of the project that resulted in the grammar, the dictionary, and the corpus. As she changed her analyses, her labels diverged from those of the original corpus—another reason for the lower accuracy seen in Table 7.1. Rather than considering the original OKMA corpus to be the ground truth, it may be helpful to view it as representing one stage in the iterative reanalysis process of language documentation.

Because each of the twelve experimental conditions (see Section 6.3.1) used examples selected from the same global pool of unlabeled examples, some duplicate clause annotation occurred for each pair of experimental conditions. Multiple labelings of a clause allow us to take simple agreement measures of both inter-annotator agreement and intra-annotator consistency.

Intra-annotator consistency. The differences between annotators also appear when we consider the consistency of each annotator’s labeling decisions. Table 7.2 and Table 7.3 (**LgExp** and **LgNov**, respectively) show, for each pair of experimental conditions, the percentage of morphemes labeled consistently by that annotator. **LgExp**’s overall average percent agreement (88.38%) is higher than **LgNov**’s (81.64%), suggesting that she maintained a more consistent mental model of the language, but one that disagrees in some areas with the OKMA annotations.

LgExp \ LgExp	seq-ns	rand-ns	unc-ns	seq-ds	rand-ds	unc-ds
seq-ns	—					
rand-ns	95.00% (41)	—				
unc-ns	87.10% (56)	90.91% (57)	—			
seq-ds	92.39% (60)	87.57% (35)	81.35% (41)	—		
rand-ds	91.02% (28)	90.94% (50)	89.10% (46)	86.13% (42)	—	
unc-ds	88.83% (51)	89.53% (57)	87.82% (332)	82.14% (42)	87.06% (49)	—

Table 7.2: Intra-annotator consistency, language expert

LgNov \ LgNov	seq-ns	rand-ns	unc-ns	seq-ds	rand-ds	unc-ds
seq-ns	—					
rand-ns	90.11% (49)	—				
unc-ns	80.80% (44)	81.68% (54)	—			
seq-ds	90.00% (54)	87.94% (44)	77.97% (48)	—		
rand-ds	90.15% (52)	86.64% (45)	79.46% (62)	81.43% (44)	—	
unc-ds	84.15% (47)	78.55% (52)	77.68% (328)	78.81% (35)	77.95% (60)	—

Table 7.3: Intra-annotator consistency, language novice

7.2 Annotator agreement, with error analysis

To get a more complete picture of the effectiveness of different levels of machine support, it is important to evaluate each annotator’s accuracy not only against the OKMA annotations, and against themselves, but also against each other. Again, the twelve experimental conditions select examples from the same global pool. The resulting clause duplication allows us to calculate simple agreement measures for both intra-annotator consistency and inter-annotator agreement.

Inter-annotator agreement. Table 7.4 shows agreement between annotators, measured in percent agreement on morphemes in clauses labeled by both annotators. The column headings refer to **LgExp**’s experiments, the

LgNov \ LgExp	seq-ns	rand-ns	unc-ns	seq-ds	rand-ds	unc-ds
seq-ns	69.91% (523)	70.82% (42)	62.42% (48)	72.35% (54)	74.25% (28)	67.82% (47)
rand-ns	71.32% (48)	83.94% (39)	66.56% (47)	66.15% (43)	73.75% (42)	67.55% (52)
unc-ns	66.31% (48)	67.87% (53)	62.31% (301)	58.87% (51)	73.31% (40)	61.10% (298)
seq-ds	73.35% (60)	75.56% (34)	56.39% (37)	60.02% (540)	66.00% (44)	61.01% (36)
rand-ds	68.67% (50)	76.40% (63)	66.67% (58)	65.88% (47)	76.33% (42)	66.99% (64)
unc-ds	65.41% (50)	67.98% (55)	60.43% (263)	58.13% (38)	70.74% (57)	60.40% (275)

Table 7.4: IAA: **LgExp** v. **LgNov**, percentage of morphemes in agreement, (number of duplicate clauses)

row headings refer to **LgNov**'s, and the number in parentheses is the number of duplicate clauses for that pair of annotator-selection-suggestion conditions. The overall average inter-annotator agreement for duplicate clauses was 66.56%. This is another indicator of the divergence from the OKMA standard analyses noted in Section 7.1, for **LgExp** in particular.

Note that the sets of clauses selected for the four pairings of uncertainty selection cases show a high level of duplication. Not surprisingly, the level of agreement for the **unc-unc** pairs is consistently well below the overall average agreement, with an average agreement of just 61.06%. This finding supports the expected result that uncertainty-based selection selects clauses that are more difficult for human annotators to label.

7.2.1 The ESP error

Another significant source of divergence for the **LgExp** from the OKMA annotations arises from instances of one of the 69 available tags. During the clean-up process, the label **ESP** was introduced for labeling Spanish loans or insertions (e.g. adverb/discourse marker *entonces* “then, in that case”). It

gradually became clear that such tokens are very inconsistently labeled in the original corpus, usually with catch-all categories such as particle or adverb. For example, the Spanish loan *nomás* “only” (segmented here as *no-mas*) often seems to function as an adverb in Uspanteko clauses (e.g. (12); text 057, clause 209).

(12) *jii'n kila' qe nomas*

Sí allí nada más.

(Yes, only there.)

In this case, the OKMA standard glosses both morphemes as adverbs (**ADV-ADV**), the **LgExp** labels the word as **ESP-ESP**, and the **LgNov** provides a split labeling (**NEG-ADV**), attempting to capture the function of each morpheme. Arguments can be made for each of the three labelings.

Further analysis is needed to determine the role such words play in the clauses they appear in: are they the product of code-switching? Do they participate in the syntax of Uspanteko? Because this question remains unresolved, and in order not to influence the predictions of the machine learner with spurious label assignments, the decision was made to mark the tokens simply as being of Spanish origin. Example (13) (text 068, phrase 110) contains two examples of this type of word—*tonses* and *pwes*—along with each annotator’s labels for the clause.

(13) TEXT: tenses wiyn pwes in ajnuch' na+

MORPH: tenses wiyn pwes in aj-nuch' na
LgExp ESP ??? ESP A1S ???-ADJ PART
LgNov ADV EXS ADV PRON GNT-ADJ PART

TRANS: *Entonces yo pues hera pequeña.*

(‘Well, I was young then.’)

Over time, the two annotators developed very different conventions for using the ESP label. The **LgExp** applied it to 2086 of 24129 tokens (8.65%) and the **LgNov** applied it to only 221 of 22819 tokens (0.97%). Because the label was introduced well after the OKMA corpus was completed, it does not appear at all in the original annotations, so any token labeled ESP is scored as incorrect when compared to the OKMA annotations; this alone adds more than seven to the **LgExp**'s total label error.

7.3 Reflections on the annotation experience

Glossing the Usbanteko texts is a tagging task. In that respect the annotators had the usual role of providing labels for items proffered for annotation. However, in these experiments annotation occurs in coordination with machine learning. In some settings the items to be labeled were selected by the machine, guided by the previously supplied labels. So, in the active learning (**unc**) cases, the labels provided by the annotator affect which examples are selected; in this way, the annotator and machine labeler are tightly coupled. Here, we consider the utility of the annotation tool and the semi-automated

annotation process from the perspective of the annotators.

7.3.1 Annotation tool

Folding machine learning into an annotation tool raises some interesting issues. For example, when offering label suggestions to the annotators, the OpenNLP IGT Editor presents the suggested labels in a separate list, as seen in Figure 6.2, but removes those labels from the alphabetically-ordered drop-down bank of possible labels. Both annotators commented that the resultant change in the ordering of the labels at times slowed down the labeling process, as they could not rely on their memory of the position of the labels within the drop-down bank.

Other issues were raised by the facts that the tool was limited to handling one stage of the process of producing IGT *and* that the tool was designed for specific experimental purposes. This restriction forces the annotators to accept the morphological segmentation as offered. The one concession made in the tool design was to offer a checkbox for flagging examples that needed further examination. The most common reason for flagging was to mark clauses with segmentation errors. In order to get accurate time measurements for labeling, it was necessary to cut out any additional analytical tasks, but in a working documentation project, this feature would likely hamper the efficiency of the annotators. Both annotators also noted that they would like to have access to the lexical gloss for stems (i.e. the stem translation) as well as the part-of-speech labels. These limitations are perhaps the main obstacles to this

tool being useful in the early stages of a documentation project.

7.3.2 Labeling-retraining cycle

Active learning is inherently cyclical: (1) a model is trained, (2) examples are selected, (3) examples are labeled; (1') the model is retrained, and so forth. In simulation studies, steps (1) and (2) tend to be time- and compute-intensive, and step (3) is trivial. This changes of course when we use real annotators, when step (3) becomes the most time-consuming step of the process. There is, however, still a time cost associated with steps (1) and (2), and the annotator generally has to wait while those steps are completed and a new batch of examples is selected for labeling. This lag time may cause frustration, distraction, boredom, or perhaps a welcome break for the annotator.

In addition, waiting time needs to be treated as part of the time cost of annotation. We did not take this into account in our experiments. However, aspects of both the experimental design and the implementation of the annotation tool combine such that annotator lag time is nearly constant across annotators and across experimental conditions, thus minimizing the impact of waiting time on our results.

First, for each annotator we alternate between experimental conditions, in order to mitigate the effect of the annotator's learning curve. Each selection-suggestion strategy combination is set up as a separate experiment, and examples are selected in batches of 10 clauses. In each round of annotation, the annotator labels a total of 60 clauses, 10 for each experimental condition.

The annotation tool is designed to work on one experiment at a time, so to switch experiments the annotator restarts the tool and is prompted to select the desired experiment. (Note that the annotators were not explicitly shown the selection method being used in each set.) Thus, switching time is consistent across experiments. Second, our models are simple, and the training set consists of only those clauses already labeled by the annotator, so the models train quickly.

Steps (1) and (2) can occur either immediately before or immediately after the batch of clauses has been labeled, and the sequence is determined by the annotator. This provides the annotator at least a small amount of control, so he/she can either proceed directly to the next experiment or wait out the short training time before switching. Also, due to the order of the steps, the model training feels more like a part of the active switching process and less like passive time sitting and waiting for the machine to finish.

7.3.3 Iterative model development

With a setup that gives annotators access to the predictions of the classifier, it is important to ask to what extent the annotators are influenced by seeing those predictions. Here we found quite different responses from the two annotators.

The **LgExp** noted that the machine’s accuracy seemed to improve over time, and that bad suggestions from the machine sometimes slowed her down, as she had to wade through a number of wrong labels to get to the label she

wanted. She also noted that at some points she found herself accepting the machine's suggested label in the case of homophonous morphemes and later rethought the label, though too late to make any changes. In other words, the appearance in the suggestion list of one of the two or more possible labels for a morpheme in some sense put the other possible choices out of mind. Once she noticed this happening, she started taking more care with such cases. Note that these are precisely the kinds of cases for which the model needs additional training data to learn to distinguish the two different analyses for the morpheme. Such a conspiracy between the annotator and the model can easily push the model off track.

The **LgNov** had a more complex relationship with the machine learner. Near the beginning of the annotation process, seeing the machine labels was actually a hindrance, compared to the no-suggest cases, in which the **LgNov** was shown the labels he had previously assigned to the given morphemes. This being a hindrance is a function of the annotator's own learning process. In the beginning, he spent quite a lot of time selecting a label for each morpheme, consulting the dictionary extensively and thinking a lot about the likely role of the morpheme. In other words, he was deeply engaged in linguistic analysis. Thus he trusted the labels *he* had previously chosen, but did a lot of second-guessing and rechecking of the suggestions made by the machine. In the future, it would be helpful to highlight machine suggestions when they correspond to labels seen with previous occurrences of the morpheme. It might also be useful to show annotators the model's level of confidence for each suggested label.

Later in the annotation process, as the model began to make more accurate predictions more consistently, **LgNov** began to trust the machine suggestions much more, provided they were consistent with his own current mental syntactic model for the language. Once he trusted the machine labels to a greater extent, having access to them saved a lot of time by reducing (often to zero) the number of clicks required to select the desired label. Interestingly, **LgNov** grew to be quite aware of the varied model accuracy in the different experimental settings. Though he didn't know this at the time, the model which felt most accurate to him during annotation was in fact his best-performing model (random selection with machine labels).

7.3.4 Epiphany effect

Without having knowledge of the accuracy of the models trained on his labels, the **LgNov** commented on having several points of 'epiphany' after which he had an easier time with the annotation. These were points at which he resolved his analysis of some frequently-occurring aspect of linguistic analysis, and these discoveries show up as bumps in graphs charting the performance of the models trained on his data.

LgNov found it hard to keep track of all the changes he was making to his mental model of the Uspanteko grammar as annotation proceeded. It appeared to him that some of the periods where it seemed the machine was slipping could have in fact been cases of it no longer matching his analysis. Also, he did not know how long it would take for the machine's predictions to

stabilize after changing his analysis of something. Would it weight his later tags greater than his earlier tags? Would an erroneous analysis early on mean it would take a while for the machine to amass enough correctly glossed tokens of such a morpheme to outweigh all of the incorrectly glossed tokens? Clearly, it would be useful to have some transparency in terms of the history of analysis of certain morphemes or constructions and also the ability to explain why a model is making a decision one way or another.

7.3.5 Handling changes in analysis

Language documentation involves both preserving examples of a language in use and discovering the nature of the language through ongoing linguistic analysis; the process does not at all fit a pipeline model. Both annotators noted changes in their analyses of particular phenomena as they proceeded with annotation. In some cases, a jump in model accuracy followed an epiphany in the annotator's own model of the language.

A deficiency of our annotation tool, and indeed a challenge for most current tools used to aid production of IGT, is that it does not allow the annotator to reannotate previous clauses as the analysis changes. One possible approach would be to couple global search (i.e. search of the entire previously-annotated corpus) with a reannotation function. This would allow an annotator to view a concordance of clauses containing the morpheme in question and to pick and choose which of the labels should be changed.

One such example concerns the morphemes *li* and *ri*. Both function

sometimes like prepositions and sometimes like demonstratives. **LgNov** began the experiment glossing all instances of both morphemes as prepositions. At some point he switched to labeling them all as demonstratives, and finally, after about 30 rounds of annotation, he began to distinguish the two functions. **LgExp** also noticed an increase over time in her accuracy and consistency in labeling these two morphemes.

Chapter 8

Conclusion

Languages are dying at an alarming rate, and documentation efforts cannot keep up without both additional resources and ways of increasing the efficacy of expended effort. Computational linguistics has the potential to support the work of documenting endangered languages. This thesis has investigated the potential of automated language processing to reduce the time cost of annotation, specifically in the context of documenting and describing endangered languages.

Contributions and results

This dissertation contributes to active learning research as well as to work on semi-automated annotation. As stated in Chapter 1, the key contributions made by this thesis are the following:

- In-depth studies of the efficacy of active learning and semi-automated annotation using actual human annotators.
- Investigation of the potential benefits of these two types of machine support for language documentation, including discussion of implemen-

tational concerns as well as annotator interactions with machine predictions and learner-guided selection.

- An annotation tool for managing the interactions between the human annotator, the machine learner, and the process of selecting examples for annotation using different selection methods.
- A brief case study of corpus clean-up and transformation. In this process, a language documentation corpus was made more consistent as well as being prepared for machine analysis.
- A general, flexible XML format for storing/representing interlinear glossed text (IGT), an important linguistic data structure.

We investigate two different strategies for reducing annotation cost: semi-automatic annotation and active learning. Section 1.1 presented the main research questions and two theses related to the research objectives. The two theses and relevant findings are presented below.

Reducing annotation cost via semi-automatic annotation

Thesis one. *Supervised machine learning can be effectively used for semi-automated annotation even when very little data is available for model development.*

One over-arching theme of our results is that machine support *can* reduce annotation cost, but achieving the potential gains requires attention to

previously-neglected human factors.

Finding one: The annotator must be modeled. The effectiveness of semi-automated annotation via machine label suggestions varies by annotator, and annotator expertise seems to be a contributing factor.

Finding two: Implementation and user interface influence effectiveness of machine support. When providing machine support to manual annotation, implementation of the annotation interface must be handled carefully. For example, time spent navigating an inefficient interface increases the time cost of annotating an individual instance, thereby reducing gains seen from machine support.

Active learning with live annotation

Thesis two. *Simulated active learning in its current state fails to model crucial human factors inherent to using AL in real annotation settings.*

Simulation studies have shown that large performance gains can be had by using learner-guided selection to produce labeled training data. These same gains cannot be expected in non-simulated annotation contexts without consideration of additional human factors.

Finding three: True annotation cost must be reflected. To accurately gauge the effectiveness of AL requires a cost measurement strategy that faithfully reflects actual annotation cost.

Finding four: Annotation context influences annotation cost. The cost to annotate a given instance is not a static value. Annotation cost is meaningful only in context and is not invariant to permutations in the ordering of examples presented to the annotator for labeling.

Finding five: The annotator must be modeled. The gains from using learner-guided sample selection instead of other selection methods depend on characteristics of the individual annotator. In particular, the annotator’s level of experience and expertise with the annotation task influences the relative effectiveness of different strategies. Thus it is important to model the individual oracle or annotator when choosing a selection strategy.

Looking ahead

The machine support strategies we have investigated show promise for speeding up production of IGT, but there are many open questions. Before these approaches (including the annotation tool we produced) are ready for use by language documentation projects, more careful attention has to be paid to implementation and interface design. This also holds more generally for active learning; a recent survey suggests that uncertainty over the effectiveness of

active learning prevents some people from incorporating user-guided selection in their annotation projects (Tomanek and Olsson, 2009).

A fundamental obstacle in live active learning is the lack of data for evaluation of the model’s performance. One possible strategy for effectively employing active learning in live annotation may be to explore a multiple-annotator setup. The results from online comparison of different annotators’ labels may be helpful for optimizing selection strategies moving forward, suiting the selection strategy to the characteristics of the individual annotator as well as the point in the annotation process. For related work, see Donmez and Carbonell (2008) and Arora et al. (2009).

The findings presented above are only the beginning, we believe, of the things to be learned from these studies and the resulting data. Through the course of the experiments, we recorded every annotation made by each annotator, in all six experimental conditions, for 58 rounds of annotation. This is a potentially rich dataset for empirical testing of theoretical results from active learning. With a bit of additional curation, for example, the data might be used to test theories of cost estimation for cost-conscious selection in active learning.

Another example: we know that each annotator labeled a good number of duplicate clauses, but we do not know how that duplication influenced ongoing annotation. The first and most natural question is to wonder whether such duplicated clauses cause an artificial raise in performance over time, as we might expect clauses to be labeled faster and with higher accuracy the second

time around. This conclusion can hold only if the annotator is consistent in her or his labeling of the morphemes in the clause. From intra-annotator consistency figures, though, we see very many cases of the same clause *not* being labeled the same way twice. And a cursory analysis of those cases shows no clear directionality. In other words, annotators seemed to change from a right answer to a wrong answer nearly as often as they changed from a wrong answer to a right answer.

The studies also raise some interesting machine learning questions. First, the results presented in Section 6.3.3 show self-training providing a strong benefit for part-of-speech tagging of previously-unseen morphemes. This merits further exploration. Second, the changing nature of the linguistic analyses in the documentation process result in a learning problem with a changing target. Machine learning approaches to concept drift (Schlimmer and Granger, 1986; Widmer, 1997) may be appropriate for this scenario.

Computational linguistics and language documentation

Most work in computational linguistics focuses on well-studied languages and languages for which digitized data is relatively abundant and readily accessible. However, less-studied languages and the difficulties encountered in documenting them present opportunities and challenges to computational linguistics. Digital data from projects documenting endangered languages offer the possibility of working with a wide range of languages, most of them typologically distinct from the usual languages of interest. New languages may

inspire new approaches to automated linguistic analysis and natural language processing. They raise the question of whether current learning models transfer to typologically-different languages. And they demand that we deal with a scarcity of data.

Challenges. It is inherent to language documentation that digitally-available data is scarce, and annotated data even scarcer. The small data problem is certainly not unique to this context, but it is specific to the situation that the small amount of data available may be all we will **ever** have. Thus we are challenged both to make creative use of the available data and to leverage any connections to external resources. For most documentation projects, annotation resources are highly limited. This fact raises the challenge to seek maximum efficiency and efficacy in annotation, as well as to reduce dependence on supervised methods.

In language documentation, annotation is itself a process of discovery (see Section 2.3). Linguistic analyses develop over the course of the project, and early annotations may disagree with later annotations. It is difficult to say at what point the analysis is finished,¹ and even more difficult to identify at what point annotation practices may have changed, unless that information has been recorded by the annotators. The challenges here are dealing with annotation noise and working with an inconsistent ‘gold standard’ dataset.

¹This question is discussed in a post and subsequent comments on the blog ‘Transient Languages and Cultures.’ Titled ‘When is a linguist’s work done and dusted?’, the post was made by Peter Austin on April 20, 2007. <http://blogs.usyd.edu.au/elac>

Opportunities One obvious motivation to work in the language documentation context is the potential to support and perhaps speed up the inherently time- and resource-sensitive work of documenting the endangered languages of the world. More importantly, working in the context of less-studied languages introduces new data sources and types. Computational linguistics can expand the range of languages—and thus the range of linguistic phenomena—it treats.

In closing, I would point out that there is a huge gap between many of the low-level language processing needs of documentary linguists and the research interests of most computational linguists. This gap is a real obstacle to fruitful cross-disciplinary research, as it presents serious challenges to both parties—and I'm not referring to the interesting sorts of challenges. I have no prescription for success. I do, however, feel there is potential for shrinking the gap, probably one small bit at a time.

Bibliography

- Arora, Shilpa, Eric H. Nyberg, and Carolyn P. Rose. 2009. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL-HLT 2009 Workshop on Active Learning for Natural Language Processing*. Boulder, CO.
- Austin, Peter K. 2006. Data and language documentation. In J. Gippert, N. P. Himmelmann, and U. Mosel, eds., *Essentials of language documentation*, pages 87–112. Mouton de Gruyter.
- Baldrige, Jason and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of Empirical Approaches to Natural Language Processing (EMNLP)*.
- Baldrige, Jason and Miles Osborne. 2008. Active learning and logarithmic opinion pools for HPSG parse selection. *Natural Language Engineering* 14(2):199–222.
- Baldrige, Jason and Alexis Palmer. 2009. How well does active learning *actually* work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore.

- Beermann, Dorothee and Pavel Mihaylov. 2009. Interlinear glossing and its role in theoretical and descriptive studies of african and other lesser-documented languages. In *Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages*.
- Bender, Emily M. 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X Conference: Computational Linguistics for Less-Studied Languages*. CSLI Publications ONLINE.
- Bender, Emily M. and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of IJCNLP-05*.
- Bender, Emily M., Dan Flickinger, Jeff Good, and Ivan A. Sag. 2004. Montage: Leveraging advances in grammar engineering, linguistic ontologies, and mark-up for the documentation of underdescribed languages. In *Proceedings of the Workshop on First Steps for Language Documentation of Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, LREC 2004*.
- Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*.

- Biesele, Megan, Beesa Crystal Boo, Dabe Kaece, Dam Botes Debe, Dixao Cgun, Gkao Martin Kaece, Hacky Kgami Gcao, Jafet Gcao Nqeni, Kaece Kallie N!ani, Koce Ui, Polina Riem, Tsamkxao Fanni Ui, Ui Charlie N!aici, Asa N!a'an, Dahm Ti N!a'an, Dixao Pari Kai, N!ani 'Kun, !Unnobe Morethlwa, Ukxa N!a'an, Xoan N!a'an, Catherine Collett, and Taesun Moon, eds. 2009. *Ju'hoan Folktales: Transcriptions and English Translations—A Literacy Primer by and for Youth and Adults of the Ju'hoan Community*. Vancouver: Trafford First Voices.
- Bird, Steven. 2009. Last words: Natural language processing and linguistic fieldwork. *Computational Linguistics* 35(3).
- Bird, Steven and Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language* 79(3):557–582.
- Bow, Catherine, Baden Hughes, and Steven Bird. 2003. Towards a general model of interlinear text. In *Proceedings of EMELD Workshop 2003: Digitizing and Annotating Texts and Field Recordings*. LSA Institute: Lansing MI, USA.
- Brants, Thorsten and Oliver Plaehn. 2000. Interactive corpus annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation*. Athens, Greece.
- Buchholz, Sabine and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on*

- Computational Natural Language Learning (CoNLL-X)*, pages 149–164. New York City: Association for Computational Linguistics.
- Can Pixabaj, Telma, Miguel Angel Vicente Méndez, María Vicente Méndez, and Oswaldo Ajcot Damián. 2007a. Text collections in Four Mayan Languages. Archived in The Archive of the Indigenous Languages of Latin America.
- Can Pixabaj, Telma Angelina, Oxla juuj Keej Maya’ Ajtz’iib’ (Group) Staff, and Centro Educativo y Cultural Maya Staff. 2007b. *Jkemiix yalaj li uspan-teko*. Guatemala: Cholsamaj Fundacion.
- Cohn, David A., Zoubin Ghahramani, and Michael I. Jordan. 1995. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, eds., *Advances in Neural Information Processing Systems*, vol. 7, pages 705–712. The MIT Press.
- Copestake, Ann. 2001. *Implementing Typed Feature Structure Grammars*. CSLI Lecture Notes.
- Creutz, M. and K. Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.* 4(1):3.
- Crystal, David. 2000. *Language Death*. Cambridge: Cambridge University Press.

- Curran, James R. and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Conference of the European Association for Computational Linguistics*, pages 91–98.
- Dobrin, Lise M., Peter K. Austin, and David Nathan. 2009. Dying to be counted: The commodification of endangered languages in documentary linguistics. *Language Documentation and Description* 6:37–52.
- Donmez, Pinar and Jaime G. Carbonell. 2008. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of CIKM08*. Napa Valley, CA.
- Farrar, Scott. 2007. *Ontolinguistics: How ontological status shapes the linguistic coding of concepts*, chap. Using ‘Ontolinguistics’ for language description. Mouton de Gruyter.
- Farrar, Scott and D. Terence Langendoen. 2003. A linguistic ontology for the semantic web. *GLOT International* 7(3):97–100.
- Gippert, Jost, Nikolaus P. Himmelmann, and Ulrike Mosel, eds. 2006. *Essentials of language documentation*. Mouton de Gruyter.
- Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27(2):153–189.
- Hachey, Ben, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *Proceedings of the 9th Conference on Computational Natural Language Learning*. Ann Arbor, MI.

- Haertel, Robbie, Eric Ringger, Kevin Seppi, James Carroll, and McClanahan Peter. 2008a. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 65–68. Columbus, Ohio: Association for Computational Linguistics.
- Haertel, Robbie A., Kevin D. Seppi, Eric K. Ringger, and James L. Carroll. 2008b. Return on investment for active learning. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*. ACL Press.
- Hale, Kenneth, Colette Craig, Nora England, LaVerne Jeanne, Michael Krauss, Lucille Watahomigie, and Akira Yamamoto. 1992. Endangered languages. *Language* 68(1):1–42.
- Himmelman, Nikolaus P. 1998. Documentary and descriptive linguistics. *Linguistics* 36:161–195.
- Himmelman, Nikolaus P. 2008. Reproduction and preservation of linguistic knowledge: Linguistics’ response to language endangerment. *Annual Review of Anthropology* 37:337–350.
- Hughes, Baden, Steven Bird, and Catherine Bow. 2003. Encoding and presenting interlinear text using xml technologies. In A. Knott and D. Estival, eds., *Proceedings of the Australasian Language Technology Workshop*.
- Leidner, Jochen L. 2007. Resource monitoring in information extraction. In *Proceedings of SIGIR 2007*.

- Lewis, William and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of IJCNLP-2008*. Hyderabad, India.
- Lewis, William D. 2006. ODIN: A model for adapting and enriching legacy infrastructure. In *Proceedings of the e-Humanities Workshop*.
- Lowe, John, Michel Jacobson, and Boyd Michailovsky. 2004. Interlinear Text Editor demonstration and Projet Archivage progress report. In *4th EMELD workshop on Linguistic Databases and Best Practice*. Detroit, MI.
- Malone, D.L. 2003. Developing curriculum materials for endangered language education: Lessons from the field. *International Journal of Bilingual Education and Bilingualism* 6(5):332–348.
- Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational linguistics* 19:313–330.
- Melville, Prem, Maytal Saar-Tsechansky, Foster Provost, and Raymond J. Mooney. 2004. Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining*.
- Moon, T. and K. Erk. 2008. Minimally supervised lemmatization scheme induction through bilingual parallel corpora. In *Proceedings of the International Conference on Global Interoperability for Language Resources*.

- Moon, Taesun, Katrin Erk, and Jason Baldridge. 2009. Unsupervised morphological segmentation and clustering with document boundaries. In *Proceedings of EMNLP-2009*.
- Ngai, Grace and David Yarowsky. 2000. Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 117–125. Hong Kong.
- Palmer, Alexis and Katrin Erk. 2007. IGT-XML: An XML format for interlinearized glossed text. In *Proceedings of the Linguistic Annotation Workshop (LAW-07), ACL07*. Prague.
- Palmer, Alexis, Taesun Moon, and Jason Baldridge. 2009. Evaluating automation strategies in language documentation. In *Proceedings of the NAACL-HLT 2009 Workshop on Active Learning for Natural Language Processing*. Boulder, CO.
- Palmer, Alexis, Taesun Moon, Jason Baldridge, Katrin Erk, Eric Campbell, and Telma Can. submitted. Computational strategies for reducing annotation effort in language documentation: A case study in creating interlinear texts for uspanteko. *Linguistic Issues in Language Technology* .
- Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago University Press.

- Poon, H., C. Cherry, and K. Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217.
- Ratnaparkhi, Adwait. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
- Richards, Michael. 2003. *Atlas lingüístico de Guatemala*. Guatemala: Servipresna, S.A.
- Ringger, Eric, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*.
- Ritz, Christian and Jens Carl Streibig. 2008. *Nonlinear Regression with R*. Springer.
- Robinson, Stuart, Greg Aumann, and Steven Bird. 2007. Managing fieldwork data with Toolbox and the Natural Language Toolkit. *Language Documentation and Conservation* 1:44–57.
- Schlimmer, Jeffrey C. and Richard H. Granger, Jr. 1986. Incremental learning from noisy data. *Machine Learning* 1(3):317–354.

- Schroeter, Ronald and Nicholas Thieberger. 2006. EOPAS, the EthnoER online representation of interlinear text. In *Proceedings of Sustainable Data from Digital Fieldwork*. University of Sydney: Sydney University Press.
- Settles, Burr. 2009. Active learning literature survey. Tech. Rep. Computer Sciences Technical Report 1648, University of Wisconsin-Madison.
- Settles, Burr and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of EMNLP 1008*. ACL Press.
- Settles, Burr, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1069–1078. ACL Press.
- Seung, H. S., Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Computational Learning Theory*, pages 287–294.
- Snyder, B. and R. Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *ACL '08*.
- Stiles, Dawn B. 1997. Four successful indigenous language programs. *Teaching Indigenous Languages* .
- Tomanek, Katrin and Udo Hahn. 2009. Semi-supervised active learning for sequence labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

- Tomanek, Katrin and Fredrik Olsson. 2009. A Web Survey on the Use of Active learning to support annotation of text data. In *Proceedings of workshop on Active Learning for NLP, NAACL HLT 2009*. Boulder, CO.
- UNESCO. 2003a. Convention for the safeguarding of the intangible cultural heritage. Tech. rep., Division of Cultural Objects and Intangible Heritage. Text available at <http://www.unesco.org/culture/ich/index.php?pg=00006>.
- UNESCO. 2003b. Language vitality and endangerment. Tech. rep., Ad Hoc Expert Group on Endangered Languages.
- Vicente Méndez, Miguel Ángel. 2007. *Diccionario bilingüe Uspanteko-Español. Cholaj Tzijb'al li Uspanteko*. Guatemala: Okma y Cholsamaj.
- Widmer, Gerhard. 1997. Tracking context changes through meta-learning. *Machine Learning* 27(3):259–286.
- Woodbury, Anthony. 2006. On thick translation in linguistic documentation. *Language documentation and description* 4.
- Woodbury, Tony. 2003. Defining documentary linguistics. *Language documentation and description* 1:35–51.
- Xia, Fei and William Lewis. 2007. Multilingual structural projection across interlinear text. In *Proceedings of HLT/NAACL 2007*. Rochester, NY.

Vita

Alexis Palmer was born in Flint, Michigan, daughter of Jody Saks and Michael Palmer. She attended high school in Gaylord, Michigan and enrolled at Oberlin College in August of 1990. Palmer transferred to the University of Michigan in Ann Arbor in January 1992 and received the Bachelor of Arts degree in English Languages and Literature in May 1994. Following her undergraduate education, Palmer worked in the private sector for seven years before beginning graduate studies at the University of Texas at Austin in the fall of 2001. There she joined the Department of Linguistics, where she is a member of the UT Computational Linguistics Lab.

Permanent address: Blumenstraße 47
66111 Saarbrücken
Germany

This dissertation was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.