Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, November 2009

Computational strategies for reducing annotation effort in language documentation

A case study in creating interlinear texts for Uspanteko

Alexis Palmer, Taesun Moon, Jason Baldridge, Katrin Erk, Eric Campbell, and Telma Kaan Pixabaj

Submitted, November,
Revised, November
Published by CSLI Publications

LiLT volume X, issue Y November 2009

Computational strategies for reducing annotation effort in language documentation

A case study in creating interlinear texts for Uspanteko

ALEXIS PALMER, TAESUN MOON, JASON BALDRIDGE, KATRIN ERK, ERIC CAMPBELL, AND TELMA KAAN PIXABAJ, *The University of Texas at Austin*

Abstract

With the urgent need to document the world's dying languages, it is important to explore ways to speed up language documentation efforts. One promising avenue is to use techniques from computational linguistics to automate some of the process. Here we consider unsupervised morphological segmentation and active learning for creating interlinear glossed text (IGT) for the Mayan language Uspanteko. The practical goal is to produce a totally annotated corpus that is as accurate as possible given limited time for manual annotation. We discuss results from several experiments that suggest there is indeed much promise in these methods but also show that further development is necessary to make them robustly useful for a wide range of conditions and tasks. We also provide a detailed discussion of how two documentary linguists perceived machine support in IGT production and how their annotation performance varied with different levels of machine support.

LiLT Volume X, Issue Y, November 2009.
Computational strategies for reducing annotation effort in language documentation.

1 Introduction

With languages dying at the rate of two each month (Crystal, 2000), there is an urgent need to create linguistically detailed records of endangered languages. Broadly outlined, documentation of a language begins with the development of a collaboration between a community of speakers of the language and an individual or group, either within or outside the community, interested in pursuing the work of documenting the language. The next stage is collection (by recording) of audio/video data and then transcription of the recorded data. This often includes developing an orthography for the language. Ideally, some portion of the transcribed texts is also translated into a language of broader communication, possibly followed by a stage of linguistic analysis and description. This analysis stage involves detailed, time-consuming linguistic annotation of the transcribed texts. This is the stage that we are interested in supporting with computational assistance. The resulting collection of data and analyses can then be used to create a variety of materials, including grammars, dictionaries, language teaching and learning materials, spell-checkers, websites, and other community-oriented language resources. Ideally, future access to the language data is ensured via archiving, publication, and other methods of storage and/or dissemination.

Computational linguistics can play an important role in reducing the workload in such efforts: models that learn from data can be used to speed up the documentation process and to pinpoint interesting examples. This paper details a set of computational strategies for aiding language documentation and experiments that test the effectiveness of those strategies in a realistic language documentation context. Specifically, we examine the effectiveness of a range of computational learning approaches, from unsupervised (inducing structure from raw text) to fully-supervised (learning from previous human annotation), for assisting the production of *interlinear glossed text* (IGT) for the Mayan language Uspanteko.

IGT is a multi-level format for presentation of linguistic data and analysis used in nearly all documentary work. It serves as the focal point of the interplay between analysis and documentation and greatly facilitates later exploration and analysis of the language. However, IGT annotations are time consuming to create entirely by hand, and both human and financial resources are extremely limited in this domain. Thus, language documentation presents an interesting test case for computational assistance to ensure consistency and maintainability of analyses and to speed up annotation in a real-life context of great import. There are a number of barriers and opportunities in attempting to do so. In this paper, we specifically address:

1. **Standardization and representation (section 2).** The data created by most documentation projects uses idiosyncratic formats and usually

contains errors that require considerable cleanup before they can be processed by computational tools. The lack of a single standard format for IGT means that representations and structures used by one project are unlikely to be compatible with those used by other projects, limiting the reusability of painstakingly annotated data. We discuss cleaning up an existing corpus of Uspanteko for our experiments and converting it to IGT-XML (Palmer and Erk, 2007).

2. **Analysis.** IGT involves morphological segmentation, translation of stems, understanding the contribution of individual morphemes to the meaning of the sentence, and labeling the glosses of stems and morphemes. We test whether some of these decisions can be made more efficiently with computational help for morphology and glossing.
   (a) **Morphology (section 3).** Words are segmented into their stems and affixes. We discuss unsupervised methods for identifying concatenative stems and affixes from raw texts as a preprocessing step for IGT creation (Moon and Erk, 2008, Moon et al., 2009).
   (b) **Glossing (section 4).** Stems and affixes are given labels that indicate their grammatical function. We summarize and expand on our previous work employing active learning and semi-automated labeling to reduce the cost of annotating these labels (Palmer et al., 2009, Baldridge and Palmer, 2009).
3. **Interaction of linguists with machine decisions (section 5).** We consider the influence of machine decisions on documentary linguists who are developing their own analysis of a language with computational support. We consider this for gloss labeling for Uspanteko with respect to a documentary linguist who is an expert in the language (Telma Kaan Pixabaj) and one who had no prior experience with it (Eric Campbell).

We also argue that language documentation raises interesting and unique challenges for computational linguistics and language technologies. Textual data from language documentation presents different issues, both linguistic and non-linguistic, than extensively-adjudicated data from well-studied languages. The challenges include (a) dealing with simple, but annoying, formatting problems, (b) working with a level of analysis (IGT) that is undeniably important in linguistics but is rarely considered by computational linguists (however, see Xia and Lewis (2007, 2008), Lewis and Xia (2008)), and (c) working with constantly changing hypotheses about how to appropriately analyze a language. Furthermore, though we describe documentation as consisting of several distinct stages, in reality different stages of the process overlap. This is especially true of the analysis process, which is in fact a discovery process in which morphological analysis/segmentation, morpheme labeling, and even transcription and translation each inform the other.

Additionally, language documentation provides the opportunity to work with a wider range of languages, including many that are typologically different from the languages of the most widely-used corpora in computational linguistics. A computational model based on a handful of related, dominant, or often-recycled languages does not stand up to scrutiny as well as a model that has been tested on a broad selection of the world's languages from diverse language groups. This setting also offers the chance to test computational approaches like active learning in a live annotation context with real human annotators, rather than *post hoc* on existing data sets, as it is typically done.

We report three main findings. First, basic computational skills like scripting and data management can be very effective in improving the quality and consistency of data annotated in language documentation projects, as well as increasing the data's suitability for reuse, both by humans and by machines. Second, both fully-automatic morphological segmentation and partially-automatic morpheme glossing show some promise for speeding up IGT production, if handled carefully. And third, to be effective, *any* computational support for language documentation must take into account the complex interactions between human annotators and automated analysis.

2 Data standardization and representation

We consider computational support for two tasks in creating IGT: morpheme segmentation and gloss labeling. To provide a realistic language documentation scenario, we work with a collection of existing texts from the Mayan language Uspanteko as the reference corpus for our experiments.

The original Uspanteko data contains a number of inconsistencies and incomplete annotations.[1] It is presented in a loose space-delimited format. 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. The cleaned-up corpus also is more suitable for reuse by linguists and other interested parties, particularly those lacking language-specific knowledge and/or linguistic training. The second stage of preparing the corpus was to convert the annotation to IGT-XML (Palmer and Erk, 2007), an extensible XML format for IGT that facilitates creation of tools for working with the data and helps ensure its longevity for future use. The resulting cleaned and converted corpus may support future work on language technologies for Uspanteko.

Here, we discuss high-level considerations for digital representations for language documentation, our choices and procedures for the data preparation step, division of the resulting materials for experimentation, and the hybrid

[1] In our experience, many corpora produced by language documentation projects contain similar inconsistencies and annotation gaps.

glosses we use in the semi-automated annotation experiments.[2]

2.1 Digital data management for language documentation

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. Bird and Simons (2003) is an extensive discussion of requirements for achieving interoperability and portability in language documentation. Different documentation efforts may require different types of annotation, and documentation projects currently use a number of different (often incompatible) tools and formats for managing their efforts. In other words, there is no single standard approach to the documentation workflow.

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. The main finding of our small survey is that approaches vary widely. One was at the early stage of eliciting individual lexical items. Two of five projects maintained transcription and translation tiers, but no morpheme-level glossing. Two more had digitized texts with full IGT: transcription, translation, and morpheme glossing. There was also wide variation in software used for transcription and/or glossing. Two projects used Shoebox/Toolbox[3], two used ELAN[4] (one of those in conjunction with Microsoft Excel) and the fifth used a combination of Microsoft Word and Microsoft Access.

Although methods, technologies, and formats vary widely even across this small sample, linguistic analysis and IGT production for language documentation involve a common set of tasks to accomplish and problems to resolve. The following is a list of underlying components required for text glossing and interlinearization:[5]

1. Development of an orthography for the language and a set of labels to be used for glossing.

[2] This section focuses on attaining *internal* consistency for a dataset. We do not here address achieving *externally-oriented* consistency through use of annotation or other standards, as it is outside the scope of this work. Development and use of such standards is a complex issue with an extensive literature. See, among others, Bird and Simons (2003), Farrar and Langendoen (2003), Barwick and Thieberger (2006), Farrar and Lewis (2007), and the outcomes and proceedings of the EMELD project and associated workshops (`http:\\emeld.org`). Another excellent collection of relevant resources and links can be found on the Cyberling wiki (`http:elanguage.net/cyberling09`).

[3] `http://www.sil.org/computing/shoebox`

[4] `http://www.lat-mpi.eu/tools/elan`

[5] We make no claim that this is a comprehensive, fully representative list.

2. Linguistic analysis, including segmentation of word forms, obtaining stem translations, and determining the contributions of non-stem morphemes to meaning.
3. Labeling stems and morphemes with glosses or parts-of-speech.
4. Iterative revision of the linguistic analysis, making appropriate changes to orthography, label set, segmentation, and gloss labels.
5. Checking consistency of labelings and analysis.
6. General digital management of data at various stages of annotation.

We focus our efforts on items 2, 3, and 4. 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. Even more than most standard annotation in computational linguistics, annotation in language documentation is itself a process of discovery. The pipeline model used in much of natural language processing is inappropriate here, presenting another significant challenge to the use of computational support.

2.2 OKMA Uspanteko corpus

In order to evaluate fully or partially automated analysis, existing annotations are needed for comparison to the predictions of the automated system. Our reference corpus is a set of texts (Kaan Pixabaj et al., 2007a) in the Mayan language Uspanteko. 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, and annotated as part of an OKMA Mayan language documentation project[6] and are currently accessible via the Archive of Indigenous Languages of Latin America (AILLA).[7]

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.[8] The transcribed and translated texts are like the Uspanteko sample shown below (text 068, clauses 283-287):

(1) 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.*

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 acos-*

[6] http://www.okma.org

[7] http://www.ailla.utexas.org

[8] See Table 2 for additional details. The set of texts available at AILLA varies somewhat from the set we used.

tumbrando. 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 available to us are of four different genres. Five 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, another is an advice text describing better ways for the community to protect the environment, and 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.

2.3 Interlinear Glossed Text

Interlinear glossed text is a flexible and efficient way of presenting multiple levels of linguistic analysis and can take many different forms (Bow et al., 2003). IGT in a readily-accessible format is an important resource that can be used to examine hypotheses on novel data (e.g. Xia and Lewis, 2007, 2008, Lewis and Xia, 2008). Furthermore, it can be used by educators and language activists to create curriculum material for language education and promote the survival of the language (Stiles, 1997, Malone, 2003, Biesele et al., 2009).

We focus here on a traditional four-line IGT format, with an additional project-defined fifth tier. The TEXT line shows the original text. The next two lines—MORPH and GLOSS—present a morphological segmentation of the text and morpheme-by-morpheme glosses, respectively. The gloss line typically includes both labels for grammatical morphemes (e.g. PL or COM) and translations of stems (e.g. *hablar* “to speak, to speak to” or *idioma* “language”). The fourth line (TRANS) is usually a translation of the original text. The following is an example from Uspanteko:[9]

(2) TEXT: Kita’ tinch’ab’ej laj inyolj iin

(3)	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.’)

[9] KEY: 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

In addition to the four lines described above, OKMA uses a fifth tier (POS), described as the word-class line. This line is a mix of traditional POS tags, positional labels (e.g. suffix, prefix), and broader linguistic categories like TAM for tense-aspect-mood.

The Leipzig Glossing Rules[10] are a recent movement toward a standardized system for IGT. 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. For machine-readability, we need a fixed *structured* representation of the data presented by IGT.

2.4 IGT-XML

For the purposes of electronic archiving and presentation, and in order to be amenable to computational analysis and support, it is necessary to have a machine-readable version of the corpus used by the documentation project. This involves a number of choices about formats and standardization. This section describes the IGT-XML format that we use.

The OKMA annotations were created using Shoebox/Toolbox, a widely-used tool for lexicon management and IGT creation, particularly in language documentation contexts. The custom, pre-XML 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. In addition, in Toolbox the glossaries, grammatical markers and segmentations are defined at the individual project level, and there is a learning curve of varying acclivity for an incoming linguist when learning how these are defined. The same problems with project definitions arise when using other software such as Microsoft Excel or Word.

Since hardware changes over time, and most pieces of software and operating systems rely on specific hardware to run, it is crucial to choose, for long-term storage of data, a format that does not depend on the availability of a single piece of software. In the ideal case, one would choose a data format that is human-readable as well as machine-readable, to enable future users to understand and access the data format even when all software that previously read the data has become obsolete.

For the experiments in semi-automatic and automatic analysis that we were planning to do, one central requirement of the representation format was *flexibility*, in particular the ability to add or exchange layers of annotation in a modular fashion. The format should be flexible as to which layers of annotation are present, and in which order they are added. It should also

[10] http://www.eva.mpg.de/lingua/resources/glossing-rules.php

```
<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>
```

Figure 1: Partial IGT-XML representation for two Uspanteko words. (translations: *salio entonces*; *then he left*.)

allow us to store, side by side, gold labels created by a human annotator and machine-created labels for the same layer of annotation.

XML formats fulfill all these requirements: They are human-readable as well as machine-readable, and they are independent of any particular software. We use the IGT-XML format (Palmer and Erk, 2007), a *mildly standoff* format. It uses globally unique IDs rather than XML embedding for linking annotation layers. In particular, <`morph`> and <`word`> annotations are kept separate. In this format, annotation layers can be added flexibly without any change to existing layers. Figure 1 shows an example.[11]

In its minimal form, IGT-XML has three blocks, for phrases, morphemes, and glosses, but it is extensible by further blocks, e.g., for POS-tags. It is also possible to have different types of annotation at the same linguistic level, for example manually created as well as automatically assigned POS-tags.

The flexibility afforded by IGT-XML is useful not only for managing automatic and semi-automatic analyses, but also for storing manual annotation. The structure of languages targeted in language documentation projects is usually not as well-studied as the structure of more intensely studied languages like English. Consequently, linguistic analysis of the language data is often tentative and subject to change. For this reason it is advantageous to have different layers of annotation that are not coupled tightly, such that

[11] Earlier XML formats proposed for IGT (e.g. Hughes et al. (2004), Hughes et al. (2003), Bow et al. (2003)) use representations which nest tiers of annotation one within the other. Strictly hierarchical formats such as the one introduced by Hughes et al. (2003) limit flexibility of annotation layers and are thus inconvenient for our purposes. In that model, the morphological analysis of a word is stored within the representation for that word, such that the addition of another, machine-generated morphological analysis would require changing the representation of each word. A somewhat more flexible format is introduced in Schroeter and Thieberger (2006), but that format too is largely tailored to flexibility in presentation rather than analysis.

individual layers can be exchanged without affecting others.

2.5 Normalization of OKMA annotations

The examples of Uspanteko shown so far have been perfectly segmented, perfectly labeled, and perfectly aligned. Each morpheme is assigned precisely one label; stem and affix status is consistently indicated by hyphenation (affixes take hyphens, stems do not); and the crucial Morph and Gloss tiers each contain the same number of elements. Consistency in labeling and alignment is essential for IGT data to be smoothly handled in our experiments. The original annotations are often messier than this. In this section, we discuss the data clean-up work that was undertaken by Palmer (a computational linguist) and Kaan (an Uspanteko language expert), working side-by-side. This proved to be a very effective combination of skills for a rapid and targeted effort to improve the machine-readability and consistency of the corpus.

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, even if that text is itself in an under-resourced language. 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, IGT versions of the texts are sometimes produced by annotators with varying levels of knowledge and/or expertise, both language-specific and pertaining to linguistic analysis. In our case, all of these factors together resulted in IGT which needed a lot of clean-up.

For this task, 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 without having to hunt through the entire corpus to find them.

Grouping of annotation tiers. For each clause of labeled text, there should be a text tier, a morpheme tier, a gloss tier, a word-class tier, and a translation 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 our manual review.

Correct	x-	el	-ch
Too much	x-	-el-	-ch
Too little	x	el	ch
Mixed	x-	-el	-ch

Table 1: Some hyphenation possibilities for a three morpheme word form.

Label consistency. Inconsistent labels occur reasonably frequently in language documentation annotation. Some errors are typographical (aka *typos*, 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 permutations: 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 Palmer with assistance on several points from Kaan, and a final list of 69 possible labels was agreed upon. Simple search-and-replace functions took care of correcting these 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 representation, 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-el-ch (COM-salir-DIR) could appear with many different hyphenations, some of which are shown in Table 1. These are natural considerations for computational work, but they may be utterly unimportant to the individual annotator or the documentation project. We used a combination of automatic morpheme type identification and targeted manual correction to address hyphenation errors.

Alignment of annotation tiers. It is also crucial to properly maintain links between source text morphemes and the gloss labels assigned to them. Specifically, we need to ensure that the Morph, Gloss, and POS lines 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 (4) and (5). 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.[12]

(4) Text: Non li in yolow rk'il

(5)	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 Kaan. Others were filled with a placeholder label ('???'). The version of the corpus used in the experiments described below includes 468 known morphemes labeled with '???'.[13]

Conversion to IGT-XML. Finally, once word-to-morpheme and morpheme-to-gloss alignment problems had been resolved, we converted the cleaned annotations into IGT-XML (Palmer and Erk, 2007) using the Shoebox/Toolbox interfaces provided in the Natural Language Toolkit (Robinson et al., 2007). 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 is much more useful for future corpus and computational studies. In addition, the automated clean-up process 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.

2.6 Organization of corpus and labels for experiments

The Uspanteko corpus was split into training, development, and held-out test sets as detailed in Table 2. Texts were chosen for each split to obtain balance with respect to genre and average clause length. These are small datasets, but the size is realistic for computational work on endangered languages.

The two tasks we focus on for producing IGT are word segmentation (determination of stems and affixes) and glossing each segment. Stems and af-

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

[13] There are 734 cases of the '???' label which appear in cases of unrecoverable morphemes. These are cases where the segmentation in the original corpus indicates the existence of a morpheme without indicating its identity. Such morphemes appear as '??' in the original corpus.

Section	Words	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 2: Detailed break-down of divisions in the corpus.

fixes each get a different type of gloss: the gloss of a stem is typically its translation whereas the gloss of an affix is a label indicating its grammatical role. The additional word-class line provides part-of-speech information for the stems, such as `VI` for *platicar*.

The target representation for the semi-automated annotation studies in section 4 is an additional tier which combines part-of-speech labels for stems with gloss labels for affixes and stand-alone morphemes. The main reason for choosing this representation was to separate the stem translation task (e.g. *hablar* for `cha'be`) from the glossing task. In an actual documentation project, *both* the stem translation and the part-of-speech label would be provided as part of the glossing process. However, stem translation is a much more indeterminate task, so we focus on predicting a refined set of gloss/POS labels. Example (6) repeats the clause in (4), adding this new combined tier. Stem labels are given in bold text, and affix labels in plain text.

(6) TEXT: `Non li in yolow rk'il`

(7) 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 (second line of (3)), we select that label. If what appears is a stem translation, we instead select the part-of-speech label from the next tier down (third line of (3)).

In the entire corpus, sixty-nine different labels appear in this combined tier. Table 3 shows the five most common part-of-speech labels (left) and the five most common gloss labels (right). The most common label, `S`, accounts for 11.3% of the tokens in the corpus.

We are currently working on making our cleaned up version of the corpus available. Details will be posted to the EARL project website.[14]

[14] `http://comp.ling.utexas.edu/earl`

S	noun	7167	**E3S**	sg.3p. 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 3: Most common labels and their frequencies: POS labels on the left, gloss labels on the right

3 Unsupervised preprocessing of morphology

In unsupervised learning, the machine learns from raw, unlabeled text, such as transcribed speech from a language documentation project. While the previous section considered computational support that involves no learning or prediction by the machine, this section discusses unsupervised learning of morphology. This work directly targets the IGT-creation subtask of segmenting word forms into their component stems and affixes.

In this section, we present unsupervised approaches that can serve as a preprocessing step to manual analysis. They focus on inducing the stems and affixes and producing (possibly noisy) segmentations or ranked segmentation candidates. We assume that the morphological pattern of the language itself — i.e. whether it is suffixal, prefixal, both, concatenative, templatic, etc. – has already been determined. We consider it a reasonable assumption that the linguist doing the analysis will have a good hypothesis on a language's morphological pattern at this level. Here, we deal only with languages that are suffixal, prefixal, or both.

We frame our morphology induction problem as a dual problem of (a) clustering of morphologically related word forms, and (b) segmentation of stems and affixes. These are closely related tasks where knowledge of one may benefit the other. Below, we outline two approaches that we have examined for unsupervised morpheme clustering and segmentation.

3.1 Cross-lingual projection

The first method that we present is that of Moon and Erk (2008). It uses bitexts (parallel texts) where linguistic resources are available for one of the languages, and corresponding words in the two texts are *aligned*. Word alignment in a parallel corpus is a mapping from each word to one or more corresponding words in the corresponding sentence in the other language (Brown et al., 1990). Developed in the context of machine translation, word alignment has also been used to *project* linguistic information from a source language, for which manual or automatic linguistic analyses are available (e.g. Spanish), to a target language, for which they are not (e.g. Uspanteko). See Yarowsky et al. (2001), Yarowsky and Ngai (2001), Snyder and Barzilay (2008) for approaches that specifically deal with morphology induction.

Following the lead of Yarowsky et al. (2001) and Yarowsky and Ngai (2001), we use word alignments to project lemmatization information across languages. The source language part of the parallel text is automatically analyzed with part-of-speech and lemma information. All target language word forms that have nonzero probability of occurring with a given source language lemma/POS tag pair can potentially be word forms of a common lemma. Of course, we need to filter out words that are synonyms but do not actually derive from the same lemma. So we designate as the target "pseudo-lemma" the target language word form with the highest probability of co-occurring with the source language lemma, and we remove from the set of candidate word forms all words that do not share a common prefix of length ≥ 4 characters with the target pseudo-lemma (this holds for suffixal languages; for prefixal languages, common suffixes are checked).

We applied this approach of learning lemmatization through projection to English as source language and German as target language. We used the German and English sections of the Europarl corpus (Koehn, 2005), and evaluated against the German TIGER corpus (Brants et al., 2002), which has manual lemma annotation. Evaluated *by type*, the approach achieved 77.2% precision, 87.4% recall, and an F-score of 79.1%. Evaluation *by token* yields 83.6% precision, 26.7% recall, and an F-score of 40.5%. This indicates that some high-frequency items are missed, which is to be expected, as high-frequency target items are most likely to be aligned with many different words in the source language.

3.2 Unsupervised induction of morphological clusters using document boundaries

One problem of the approach of Moon and Erk (2008)—as well as many other methods for unsupervised morphological analysis (Harris, 1955, Hafer and Weiss, 1974, Jacquemin, 1997, Gaussier, 1999, Goldsmith, 2001, Schone and Jurafsky, 2001, Freitag, 2005, Demberg, 2007)—is its reliance on multiple parameters, such as the requirement of a prefix overlap with the target pseudo-lemma of at least 4 characters. This is problematic in a documentary setting: Parameters that need manual tuning are especially bad since they place an additional workload on the documentary linguist. But even parameters that are automatically calibrated on data are problematic since usually the amounts of data available in a documentary setting are relatively small.

So the second approach that we present is that of Moon et al. (2009), which attempts to eliminate parameters as much as possible. A key idea is to exploit a feature of documentary data (as well as most corpora) that is near universal– that document boundaries are naturally preserved in these datasets. The simple intuition is that if orthographically similar words occur within the same document, there is a good chance that they are morphologically related.

The approach proceeds in four stages, again addressing the problems of clustering and segmentation discussed above. (1) The first step is a segmentation step, generating suffix candidates whenever a common stem candidate is found to occur with multiple different endings (and analogously for prefixes). The criterion for identifying stem candidates is based on the intuition that stems are longer than affixes.[15] This step overgenerates, so we (2) filter candidate affixes, retaining only those that show statistically significant co-occurrence with shared stems. To test significance, we use pairwise χ^2 tests. The remaining steps are clustering steps: After (3) clustering affixes, we (4) cluster stems based on affix clusters.

Document boundaries are utilized in stages (1) and (4). In step (1) we count stem co-occurrence either by document or globally. In step (4) we cluster stems that occur, either in the same document or globally, with affixes in the same cluster.

The model was applied to two data sets from English and to the Uspanteko data set discussed in section 2. (*Taesun*) *While English is not currently endangered and seems an inadequate subject for exploring the feasibility of this model, English is the* de facto *language in unsupervised approaches to morphology (Schone and Jurafsky, 2000, Freitag, 2005, Poon et al., 2009) and merits inclusion by strength of convention alone. It is also easy to evaluate models on, due to CELEX (Baayen et al., 1993) and the easy availability of off-the-shelf lemmatizers and stemmers. On its own, English presents interesting challanges in terms of morphological analysis due to its tendency to carry orthographical baggage from centuries past and from a grab-bag of foreign languages, baggage that does not exist for languages that only recently acquired a writing system. We were unable to include other truly endangered languages due to the difficulty of obtaining gold evaluation standards and the lack of access to native speakers of such languages who could have manually evaluated our model output.* For English, we used a larger and a smaller dataset, of 9M and 187K word tokens, respectively, to test the effect of dataset size on the model. In both cases, the data came from the *New York Times* segment of the Gigaword Corpus.[16] The best performing model on both English datasets, large and small, used global search for segmentation (step 1) and document-aware clustering (step 4). Precision/recall/F-score for the larger and smaller data were $77.7/70.2/73.8$ and $88.3/78.0/82.8$ respectively. Awareness of document boundaries seems

[15] In the case of English, there are exceptions such as *be/be-ing, do/do-ing*, but they are not prevalent as *types*.

[16] LDC catalog no.: LDC2003T05

particularly helpful with smaller corpora: On the small corpus, the completely document-aware approach (applied in steps 1+4) outperformed a completely global approach, with the opposite results on the large corpus. All versions of our model achieved higher F-scores than two benchmark systems, Linguistica (Goldsmith, 2001) (*Taesun*) *—which achieved 76.2 on the smaller and 61.8 on the larger*— and Morfessor (Creutz and Lagus, 2007) (*Taesun*) *—which achieved 59.7 on the smaller and 66.3 on the larger*.[17]

Uspanteko morphology is polysynthetic, with both productive prefixes and productive suffixes, so we tested three different assumptions with the model: that the language is only prefixal, only suffixal, and suffixal+prefixal with no concatenative morphology. The best results were achieved by a fully global model viewing Uspanteko as prefixal, with precision 92.0, recall 50.0, and an F-score of 64.8. (*Taesun*) *Linguistica and Morfessor achieved F-score of 64.3 and 38.8, respectively.*

The effectiveness of global segmentation (step 1) is due to the heuristic that assumes that stems will be longer than affixes, and thereby suppresses the generation of short stems. Short words account for most noise through spurious affixes, and with the larger, global datasets, there is a greater chance that a short word will be filtered out because it happens to have a large overlap with a longer unrelated word. Document based clustering is particularly effective when coupled with global segmentation: It blocks the clustering of potential morphological variants which have never co-occurred in a document. This boosts precision. We are still examining why on smaller data sets, document based segmentation and clustering show strong precision in spite of the fact that step (1) may generate noisy candidates.

It remains to be seen how useful the output from these models is for creating interlinear glossed texts as part of an overall language documentation process. However, manual evaluation by Kaan of the output of our model is encouraging. She measured the accuracy of 100 random morphological clusters produced by the model for Uspanteko and found individual clusters to be 98.5% accurate on average, with complete accuracy on 79.0% of all clusters. (*Taesun*) *Linguistica had accuracy of 96.0% and 85.0% full cluster accuracy. Morfessor had 85.3% accuracy and 55.0% full cluster accuracy.* See Moon et al. (2009) for more details on the model and evaluation. More extensive evaluation needs to consider two factors: (1) the effect of automatic preprocessing on annotation speed; and (2) the effect on annotation consistency and correctness.

Another possible extension concerns the question of pipeline-style processing versus integrated models. In the process of IGT creation, we are cur-

[17] Note that both were used with default parameters and that Morfessor was not used for its intended purpose of pure segmentation.

rently considering morphological analysis (section 3) and part-of-speech tagging (section 4) completely separately. The two processes ideally inform each other, but it is open whether the advantages gained by this information could outweigh the added complexity of an integrated model.

4 Semi-automated annotation

The previous section discussed an *unsupervised* model for preprocessing a corpus to suggest morphological variants and analyses. In this section we discuss models that receive *supervision*, i.e., they learn from previously given human annotation. The amount of material which gets annotated is often limited by the money available for annotation. In the language documentation scenario, there is the additional hard constraint of time running out—for severely-endangered languages there may be no more than a small part of one generation still using the language, and before long there may be no more living knowledge of the language.

This section considers how a machine learning system can *interact with* an annotator to efficiently improve its accuracy such that it can be used for reliable labeling of new material. When annotation is done, we seek to have a corpus which is maximally useful for training an accurate classifier that can label further material reliably. This relies on two aspects of machine learning systems which have little supervised (i.e. human-labeled) training data: (1) not all examples are equally valuable to machine learners and (2) even when a machine learner is unsure on an example, it often assigns a high probability to the correct label, compared to probabilities for all other possible labels.

In Palmer et al. (2009) and Baldridge and Palmer (2009), we describe a series of annotation experiments designed to test the viability of exploiting these two aspects to speed up morpheme gloss labeling. The practical goal is to explore best practices for using automated support to create fully-annotated texts. The aim is to achieve the highest quality possible within fixed resource limits. Palmer et al. (2009) describes our data preparation and initial results for the use of active learning on the task of morpheme glossing. Baldridge and Palmer (2009) gives a detailed comparison of different strategies and conditions in terms of their relative effectiveness. Here, we provide a few details not covered in those papers and summarize the experiments and results.

4.1 Active learning

Active learning has one core driving principle: we should use our human experts as effectively as possible, so we should avoid asking them for labels for easy examples. Examples which present some novelty are likely to help a machine learner improve its performance more quickly. In brief, active learning attempts to maximize the impact of human annotation time by identifying informative examples for the human to annotate. 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 analyze a large set of previously unseen examples, a set of maximally-informative examples is selected from this pool, and the selected set is annotated by a human and added to the training data. The model is then retrained on the seed set plus the newly annotated examples, and the cycle repeats. With respect to the need for labeled data, active learning is well-suited for the language documentation context, in which it is common for a project to produce a small amount of IGT-annotated data and a much greater amount of unannotated data.

The active learning method we use is uncertainty sampling (Cohn et al., 1995). Uncertainty sampling identifies examples the model is least confident about. 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.

We compare **unc**ertainty selection against two baseline methods: **seq**uential and **rand**om. 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 genre), 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.

4.2 Label suggestions

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.[18] 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 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. It is also extremely likely that ranking by frequency helps considerably in determining the correct label.

4.3 Annotators and (lack of) annotation conventions

The annotations in the experiments were performed by Campbell and Kaan. Both are trained 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.

Kaan has done extensive linguistic and lexicographic work on Uspanteko. Her work includes a written grammar of the language (Kaan Pixabaj et al., 2007b) and contributions to the publication of an Uspanteko-Spanish dictionary (Vicente Méndez, 2007). Additionally, Kaan is a native speaker of K'ichee', a Mayan language that is closely related to Uspanteko.

Campbell 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, Campbell 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.

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 also (or sometimes instead) draw on the time of a linguist *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 does influence which selection strategies are most effective.[19]

A factor related to expertise is that not all annotators cost the same. For example, the most knowledgeable and possibly most efficient annotator might

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

[19] It 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.

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.

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. 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 which matched and which did not 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.

4.4 Annotation tool

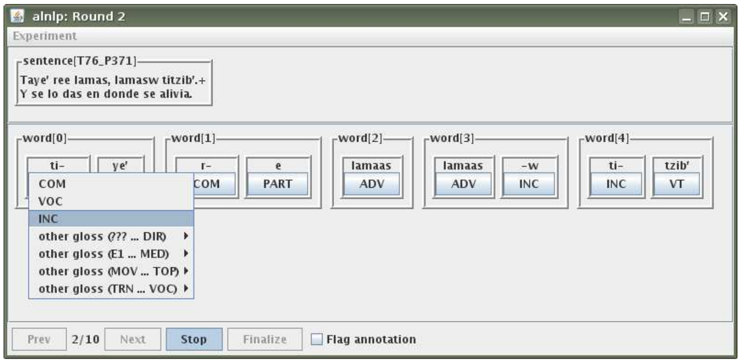
To evaluate the effectiveness of machine support in these different conditions requires integrating automated analysis into the manual annotation process. The integration in turn requires careful coordination of three components: 1) presenting examples to the annotator and storing the annotations, 2) 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 Editor[20], 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).[21]

An example of annotating a clause with the IGT editor is given in Figure 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 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

[20] `http://igt.sourceforge.net/`

[21] `http://michel.jacobson.free.fr/ITE/index_en.html`



Taye' ree lamas, lamasw titzib'.+
Y se lo das en donde se alivia.

Figure 2: The OpenNLP IGT Editor interface.

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, and 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

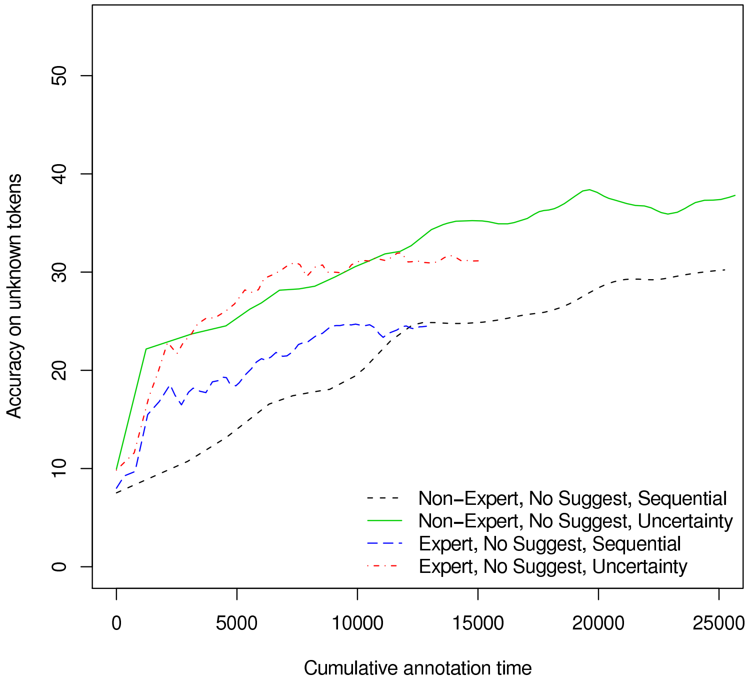


Figure 3: Accuracy on previously-unseen morphemes for both annotators, **seq** vs. **unc**.

those of the non-computationally-savvy 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 section 5.

The OpenNLP IGT Editor is available under the open source Lesser General Public License.[22] We hope that it will enable the language documentation community to take advantage of our tools and techniques.

4.5 Findings

One of the biggest findings—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 Campbell (the annotator without Uspanteko language expertise) was the most effective annotator, but this result reversed

[22] `http://www.gnu.org/licenses/lgpl.html`

when the time used to annotate was taken into account: with time cost, Kaan produced datasets that trained more accurate classifiers much more quickly.

The second, more surprising, finding is that uncertainty selection worked well with Kaan, but it performed worse than random selection with Campbell. 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 (although see (Donmez and Carbonell, 2008) and (Arora et al., 2009)), yet it is clearly essential if active learning is to be an effective technique in real-life annotation projects.

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 3 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 analyze the long tail of words/constructions in the language that may not be common but are linguistically interesting. Here uncertainty selection shines in our experiments for both annotators in all conditions.

The third major finding is that label suggestions provided by the machine were useful for Campbell but not for Kaan. Campbell found the suggestions useful for limiting the likely analyses for a given morpheme, whereas Kaan 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. 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 there is 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 uncertainty regarding how well active learning works in practical applications. 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.

5 Interaction of annotators with machine decisions

A key part of our approach to reducing annotation cost is to integrate machine support and manual annotation. In this case, the annotations provided are labels for morphemes in IGT, and the annotators are two documentary linguists with extensive field experience. The annotators worked in a number of different experimental settings with different levels of machine support, which comes in the forms of both active learning and machine label suggestion. The annotators found both advantages and disadvantages to working with machine assistance. Also the two annotators differ greatly with respect to their previous experience with Uspanteko, and the different levels of expertise had a strong effect on which levels and types of machine support were most helpful.[23] These attributes also changed as annotation progressed and the annotators progressed along their own learning curve.

5.1 Learning curve

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 Kaan's annotation times line up with this typical case, Campbell's learning curve is much steeper; in addition to familiarization with guidelines and interface, Campbell is in fact discovering the nature of the language as he goes.

As expected, Kaan's learning curve reached a plateau far quicker than Campbell's. 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. Campbell, starting from zero, needed much more work to acquire proficiency with the language and task. This is reflected especially in his average annotation time per morpheme, shown in Figure 4.

Campbell noted clear patterns in his acquisition of a syntactic model for

[23] As a reminder, Kaan is the language-expert, and Campbell is the language-novice.

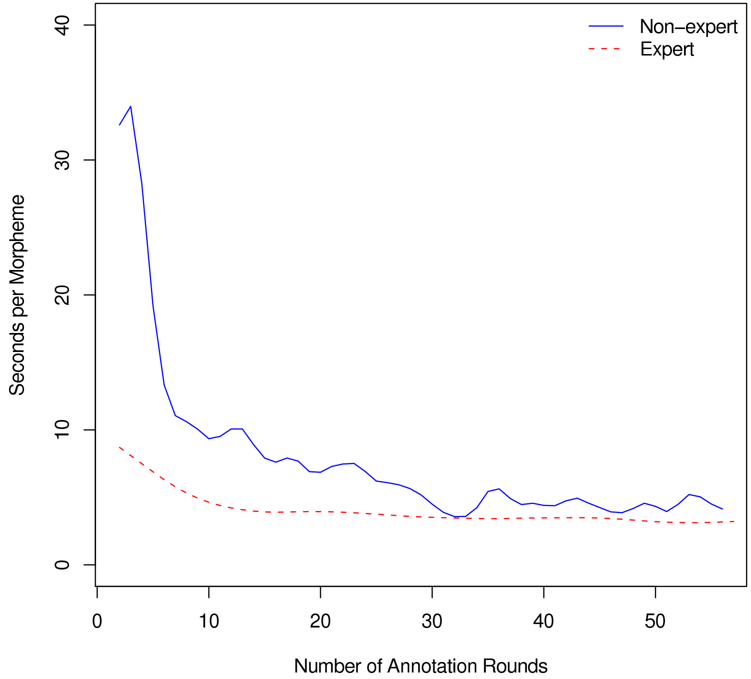


Figure 4: Average annotation time (in seconds per morpheme) over annotation rounds, averaged over all six conditions for each annotator.

the language: verbs became clear first, because in Uspanteko only verbs take aspect marking. Person marking was the key to his next major step, which was the identification of relational nouns and possessed nouns, both of which inflect with the ergative person markers (set A in traditional Mayan linguistics). Adjectives were the third major area in which he felt confident of his labels, since almost invariably they immediately precede the nouns they modify. The most difficult distinctions, in his opinion, were sentence-initial morphemes, variously labeled as adverbs, particles, or affirmatives.

A complicating factor for both annotators is that the data marks both derivational and inflectional morphology, but the former is not consistently analysed as such. In other words, in some cases the segmentations take derivational morphology into account and in others they do not.

Access to the Uspanteko-Spanish dictionary was essential for Campbell to make any progress understanding the language. In fact, he found simultaneous use of the digital and printed versions of the dictionary to be the most effective strategy. The digital version allows the use of targeted search terms, while the hard copy lets the user rapidly scan many pages of entries (and their examples) to get a broader picture of the language. At the same time, a dictionary as a stand-alone resource is of course not sufficient for developing a comprehensive knowledge of the linguistics of a language.

5.2 Annotator label accuracy

In section 4, the effectiveness of different levels of machine support was discussed with respect to the labeling accuracy of models learned from the training data produced by the annotators. Accuracy of a model is evaluated by comparing its label predictions to the original (i.e. OKMA) annotations on a set of held-out texts. Thus the potential for learning an accurate model is affected by the accuracy of the annotators' labels as compared to the original annotations.

Accuracy against OKMA annotations. Table 4 shows the overall accuracy of the annotators' labels for each condition (after 56 rounds) as measured against the original OKMA annotations. As expected, **unc** selection tends to pick examples that are more difficult to label. Accuracy for both annotators suffers in both **unc-ns** and **unc-ds**.

The fact that Campbell's accuracy is generally higher than Kaan's is initially surprising; 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; Campbell spent nearly twice as much time labeling the same number of examples. The more important factor, though, again has to do with prior experience with Uspanteko. In language documentation, the analysis of the language is continually evolving. The OKMA annotations represent less a ground truth for the language than a reflection of the understanding of Uspanteko at the time that the original annotations were done. Kaan recognized—often through the morphological segmentation shown by the annotation tool—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.[24] As a result, her labels in cases diverge from those of the original corpus.

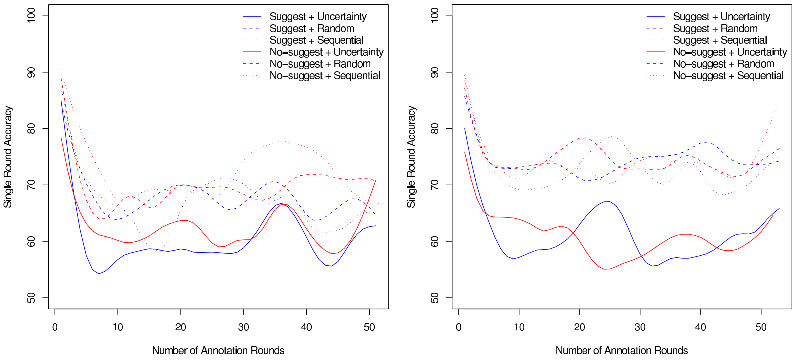
Annotator accuracy by round. Though annotator speed clearly improves over time, the same correlation does not hold for the accuracy of the annotators' labels. Figure 5 plots annotator label accuracy by round of annotation.[25] Despite showing lower label accuracy, faster annotation rates allowed Kaan to achieve higher accuracy in much less time.

[24] This is the main reason for providing the 'Flag annotation' checkbox as part of the annotation tool interface.

[25] The left terminus of each plot line is artificially high, as it plots the accuracy of the 10-clause seed set, which is drawn directly from the OKMA data.

	expert	**non-expert**
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 4: Overall accuracy of annotators' labels, measured against OKMA annotations.



(a) (b)

Figure 5: Single round accuracy per round for each experiment type by: (a) expert annotator (Kaan), (b) non-expert annotator (Campbell).

The ESP error. Another significant source of divergence for Kaan from the OKMA annotations arises from one individual label. During the clean-up process, the label ESP was introduced for labeling Spanish loans or insertions (such as the adverb/discourse marker *entonces*). It gradually became clear that such tokens are very inconsistently labeled in the original corpus, usually with catch-all categories like particle or adverb. For example, the Spanish loan *nomas* (segmented, perhaps controversially, as no-mas) often seems to function as an adverb in Uspanteko clauses (e.g. (8); text 057, clause 209).

(8) jii'n kila' qe nomas
Si alli nada màs.

In this case, the OKMA standard glosses both morphemes as adverbs (ADV), Kaan labels both with ESP, and Campbell provides a split labeling, attempting to capture the function of each morpheme: NEG-ADV. 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 (9) (text 068, phrase 110) contains two examples of this type of word—*tonses* and *pwes*—along with each annotator's labels for the clause.

(9) TEXT: tonses wiyn pwes in ajnuch' na+

MORPH:	tonses	wiyn	pwes	in	aj-nuch'	na
KAAN:	ESP	???	ESP	A1S	???-ADJ	PART
CAMP:	ADV	EXS	ADV	PRON	GNT-ADJ	PART

Entonces yo pues hera pequeña.

Over time, the two annotators developed very different conventions for using the ESP label. Kaan applied it to 2086 of 24129 tokens (8.65%) and Campbell 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 7 percentage points to Kaan's total label error.

5.3 Annotator agreement

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

exp non	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 5: Inter-annotator agreement: expert v. non-expert, percentage of morphemes in agreement, (number of duplicate clauses)

only against the OKMA annotations, but also against each other and even against themselves. Because each of the twelve settings (annotator-selection-suggestion) 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.

Inter-annotator agreement. Table 5 shows agreement between annotators, measured in percent agreement on morphemes in clauses labeled by both annotators. The column headings refer to Kaan's experiments, the row headings refer to Campbell'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 5.2, for Kaan in particular.

Note that the sets of clauses selected for the four pairings of uncertainty selection cases show a very 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 does indeed select clauses that are more difficult for human annotators to label.

Intra-annotator consistency. The differences between annotators also appear when we consider the consistency of each annotator's labeling decisions. Table 6 and Table 7 (expert and non-expert, respectively) show, for each pair of experimental conditions, the percentage of morphemes labeled consistently by that annotator. Kaan's overall average percent agreement (88.38%) is higher that Campbell'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.

exp **exp**	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 6: Intra-annotator consistency, expert annotator

non **non**	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: Intra-annotator consistency, non-expert annotator

5.4 Reflections on the annotation experience

Glossing the Uspanteko 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 cases, the annotator's labels affected which examples were 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.

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 labels in a separate list, as seen in Figure 2, but removes the suggested 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, by far, 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.

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 even a much needed 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 unaware of the specific experimental conditions for each set.) Thus, experiment-switching time is the same for all 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.

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.

Kaan 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 of one of 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. We note that these are precisely the kinds of cases for which the machine 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.

Campbell 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 Campbell 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.

Later in the annotation process, as the model began to make more accurate predictions more consistently, he began to trust the machine suggestions much more, provided they were consistent with his own current mental syntactic model for the language. Once Campbell 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, Campbell grew to be quite aware of the varied model accuracy in the different experimental settings. In fact, though he didn't know this, his impression of the most accurate model is indeed the same as his best-performing model (random selection with machine labels).

Epiphany effect. Without having knowledge of the accuracy of the models trained on his labels, Campbell 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.

Campbell found it hard to keep track of all the changes he was making all along in his mental model of the Uspanteko grammar. 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 an erroneous analysis early on mean it would 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.

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 fact, each annotator discovered a previously-unknown aspect of the morphosyntax of verbs in Uspanteko. 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 any tool 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. Campbell 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. Kaan also noticed an increase in her accuracy and consistency over time.

6 Conclusion

Based on the results of the work and experiments described in this paper, we believe there is clear potential for fruitful, mutually beneficial collaboration between language documentation and computational linguistics.

Challenges and benefits for language documentation. Documenting and describing an endangered language is a complex task with no clearly established best methodology or workflow.[26] Each language offers its own set of challenges, and each documentation project tends to develop its own set of solutions. An additional confound to consistency in the documentation process is the fact that most such projects are individual or small-group endeavors on small budgets, with little or no institutional guidance by the greater documentary linguistics community.[27]

Even very simple computational strategies, such as basic scripting for text manipulation and data management can be very effective in efficiently improving the quality and consistency of transcriptions, translations, and IGT annotations from language documentation projects. Much of the work of documentation is data-intensive and corpus-based, and many of the problems encountered could greatly benefit from knowing a scripting language such as Perl, Python or Ruby. In addition, these skills and use of standard formats both greatly increase the reusability of such data.

Machine learning and active learning approaches, while certainly more complex and more challenging to implement, show some promise for partially automating and thus speeding up the creation of IGT. Using machine assistance, we consistently learned more accurate models, more quickly, than was possible using the standard strategy of sequential annotation with no machine label suggestions. These learning methods need input data that is in a well-organized and machine-readable format, but they also output this type of data, supporting the possibility for future work on language technologies for the language being documented.

Challenges and benefits for computational linguistics. Questions arise in the context of documentary linguistics that present interesting challenges for computational linguistics. IGT creation is actually a bad fit for the standard pipeline model frequently used in computational linguistics, because the different stages of analysis—morphological analysis/segmentation, labeling of morphemes, and sometimes even translation—overlap. Another challenge is the absence of strict annotation guidelines; both analysis and annotation are

[26] Although strong recommendations exist for ensuring the longevity and robust accessibility of digitized language data; see for example Bird and Simons (2003) and the results of the EMELD project `http://emeld.org`.

[27] Some personal perspectives on the difficulties faced by such projects can be found in Newman (1992), Wilkins (1992), Rice (2001).

continually evolving, each informing the other.

Some of the challenges we encountered raise important considerations for direct practical application of machine learning and active learning. First, the interactions between the human annotator and the machine learner are very complex and must be carefully considered and thoroughly addressed for effective integration of the two. In other words, unanticipated human factors can diminish the potential effectiveness of input from the machine learner. Second, for active learning to be effective, some sort of model of the annotator needs to be incorporated into example selection strategies. Finally, it is important that annotation software be flexible enough to allow for revision as the analysis of the language changes; at the minimum, computational support can assist with propagating such changes back to previously-labeled clauses.

Research in computational linguistics also stands to benefit from expanding the range of languages it works with. Standardized, machine-readable IGT annotations for less-studied languages and the diverse phenomena they exhibit would enable a much wider cross-linguistic validation of models used in computational linguistics. Also, machine learning techniques have mostly been used in scenarios where large volumes of data are available. There is thus an opportunity to evaluate the impact of models that assume less training material, e.g. through linguistically informed priors.

Infrastructure. Creating the necessary software infrastructure to build an active learning system is a substantial hurdle. Creating our annotation tool to interface between data, annotator, and machine classifier required considerable effort, especially to ensure it was easy to use for the annotators. If machine learning is to be of any assistance to language documentation, beyond individual projects such as ours, it is clear that a basic digital infrastructure is necessary, and that this infrastructure must take into account both machine and human factors.

We envision a broadly-accessible, web-based system for collaborative annotation of texts, possibly based on the model used with Amazon's Mechanical Turk (AMT). Using AMT, labels can be obtained for a fraction of the usual cost by parceling out instances to self-selected non-expert annotators. AMT labels have been found to show high agreement with pre-determined gold-standard labels for some natural language processing tasks, such as affect recognition and recognizing textual entailment (Snow et al., 2008). By definition, though, AMT labels are non-expert labels and so not likely to be useful for work in language documentation. Nevertheless, AMT stands as an example of the potential for massive, global collaboration on tasks involving textual data. With an AMT-like system specifically targeted for IGT and involving a community of documentary linguists as annotators, the documentation process could be sped up significantly.

Though we envision this system as facilitating collaboration not just between many humans, but also between humans and machines, it is abundantly clear that the machine learner must be tuned to the needs of the human. Questions of human-computer interaction must be carefully considered, the learner must model what the human *actually* does, and the human must always retain veto power.

Acknowledgments

This work is funded by NSF grant BCS 06651988 “Reducing Annotation Effort in the Documentation of Languages using Machine Learning and Active Learning”. Documentary field work for Taesun Moon was extensively funded by the Kalahari People’s Fund. Access to the Uspanteko corpus prior to its archiving was provided by the OKMA academy of Guatemala. Thanks to Emily Bender, Nora England, Michel Jacobson, Terry Langendoen, Elias Ponvert, Tony Woodbury, the anonymous reviewers, and the participants of the organized session “Computational Linguistics for Linguistic Analysis” at the LSA 2009 annual meeting for helpful and stimulating discussion and feedback.

References

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.

Baayen, R. H., R. Piepenbrock, and Rijn van H. 1993. *The CELEX lexical data base on CD-ROM*. Philadelphia, PA: Linguistic Data Consortium.

Baldridge, 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.

Barwick, Linda and Nicholas Thieberger, eds. 2006. *Sustainable Data from Digital Fieldwork*. Sydney University Press.

Biesele, Megan, Beesa Crystal Boo, Dabe Kaqece, Dam Botes Debe, Di–xao Cgun, G″kao Martin —Kaece, Hacky Kgami Gcao, Jafet Gcao Nqeni, Kaqece Kallie N!ani, —Koce —Ui, Polina Riem, Tsamkxao Fanni —Ui, —Ui Charlie N!aici, —Asa N!a’an, Dahm Ti N!a’an, Di–xao Pari —Kai, N!ani —’Kun, !Unn—obe 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 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, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

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

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.

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.

Demberg, V. 2007. A language-independent unsupervised model for morphological segmentation. In *ACL '07*, vol. 45, page 920.

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 and D. Terence Langendoen. 2003. A linguistic ontology for the Semantic Web. *GLOT International* 7(3):97–100.

Farrar, Scott and William D. Lewis. 2007. The GOLD community of practice: An infrastructure for linguistic data on the web. *Language Resources and Evaluation* .

Freitag, D. 2005. Morphology induction from term clusters. In *CoNLL '05*.

Gaussier, E. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *ACL workshop on Unsupervised Methods in Natural Language Learning*.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27(2):153–198.

Hafer, M.A. and S.F. Weiss. 1974. Word Segmentation by Letter Successor Varieties. *Information Storage and Retrieval* 10:371–385.

Harris, Z.S. 1955. From phoneme to morpheme. *Language* 31(2):190–222.

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*, pages 105–113.

Hughes, Baden, Catherine Bow, and Steven Bird. 2004. Functional requirements for an interlinear text editor. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 771–776.

Jacquemin, Christian. 1997. Guessing morphology from terms and corpora. In *SIGIR '97*, pages 156–165. ISBN 0-89791-836-3.

Kaan 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.

Kaan Pixabaj, Telma Angelina, Oxlajuuj Keej Maya' Ajtz'iib' (Group) Staff, and Centro Educativo y Cultural Maya Staff. 2007b. *Jkemiix yalaj li uspanteko*. Guatemala: Cholsamaj Fundacion.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*.

Lewis, William and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of IJCNLP-2008*. Hyderabad, India.

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.

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*, pages 179–186.

Moon, Taesun, Katrin Erk, and Jason Baldridge. 2009. Unsupervised morphological segmentation and clustering with document boundaries. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore.

Newman, P. 1992. Fieldwork and field methods in linguistics. *California Linguistic Notes* 23(2):1–8.

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, A., T. Moon, and J. 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.

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. Association for Computational Linguistics.

Rice, K. 2001. Learning as one goes. *Linguistic fieldwork* pages 230–249.

Richards, Michael. 2003. *Atlas lingüístico de Guatemala*. Guatemala: Servipresna, S.A.

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.

Schone, P. and D. Jurafsky. 2000. Knowledge-free induction of morphology using latent sematic analysis. In *CoNLL-2000 and LLL-2000*.

Schone, Patrick and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of NAACL*, pages 1–9. Association for Computational Linguistics.

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.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP 2008*, pages 254–263.

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* .

Vicente Méndez, Miguel Ángel. 2007. *Diccionario bilingüe Uspanteko-Español. Cholaj Tzijb'al li Uspanteko*. Guatemala: Okma y Cholsamaj.

Wilkins, D. 1992. Linguistic research under Aboriginal control: A personal account of fieldwork in central Australia. *Australian Journal of Linguistics* 12(1):171–200.

Xia, Fei and William Lewis. 2007. Multilingual structural projection across interlinear text. In *Proceedings of HLT/NAACL 2007*. Rochester, NY.

Xia, Fei and William Lewis. 2008. Repurposing theoretical linguistic data for tool development and search. In *Proceedings of IJCNLP-2008*. Hyderabad, India.

Yarowsky, David and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of NAACL-2001*, pages 200–207.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of First International Conference on Human Language Technology Research (HLT 2001)*.