Applying Computational Linguistic Techniques in a Documentary Project for Q'anjob'al (Mayan, Guatemala)

Jonas Kuhn, B'alam Mateo-Toledo

The University of Texas at Austin
Department of Linguistics
Austin, TX 78712, USA
jonask@mail.utexas.edu, tbalam@mail.utexas.edu

Abstract

This paper reports on a number of experiments in which we applied standard techniques from NLP in the context of documentation of endangered languages. We concentrated on the use of existing, freely available toolkits. Specifically, we explore the use of Finite-State Morphological Analysis, Maximum Entropy Part-of-Speech Tagging, and N-Gram Language Modeling.

1. Introduction

In this paper we explore the use of techniques and tools from Computational Linguistics and Natural Language Processing (NLP) in the context of an effort to document and revitalize the Mayan language Q'anjob'al, spoken in Guatemala. Currently, there is a community of about 90,000 speakers of Q'anjob'al (Richards, 2003, 74). The available resources for the language are fairly limited; however, there are active efforts in recording, digitizing, and transcribing speech material. The goal of this ongoing project has been to find the right level of tool support that will speed up the creation of medium-size or large language resources to serve both in the revitalization efforts and in linguistic research. While quality and depth of the language resources are important criteria, quantity and ease of resource creation should not be underestimated as important factors in work with endangered languages. Although most NLP techniques require either extensive language-specific linguistic modeling or large amounts of training data, we believe that some techniques, applied interactively, can be very helpful in the documentary process. We are particularly interested in tools that can complement the standard toolkit in field linguistics, Shoebox.[1] Findings from our project will hopefully generalize to other similar efforts.

Our investigations in this paper focus on NLP techniques that are (i) relatively well-established in the field of NLP, and for which (ii) freely available toolkits or program components are available. These are important criteria for making sure that the methodology is applicable on a broader basis. After giving some further background on Q'anjob'al and our corpus for this language in section 2., we will discuss our experience in experimenting with three NLP techniques: Finite-state Morphological Analysis (section 3.), Maximum Entropy Part-of-Speech Tagging (section 4.), and N-Gram Language Modeling (section 5.). Each section includes a discussion of how useful we found the particular approach to be in our project context. Section 6. provides a short general conclusion.

2. Background

Our project is in the fortunate position that one of the authors is a native speaker of Q'anjob'al. The raw language material we have currently available are about 40 hours of recorded speech, collected by this author over the past three years, 8 hours of which have been digitized and transcribed as plain text. The collected data of transcribed speech comprise stories, songs, legends, prayers, etc. We also used some written text – a series of texts from a pedagogical grammar of about 7,500 words. In total, the corpus contains 70,000 running word forms.

Q'anjob'al is a head marking ergative language with split ergativity. The language has a strict VSO word order; the arguments are marked morphologically on the predicate head by cross-reference markers. A formal distinction between verbal and nonverbal predicates is made through aspect and person markers on the predicate head. Practically any of the major word classes and particles can appear as nonverbal predicate heads. The structure of the predicate head shows some complexity because most of what we know as inflectional categories (except person markers) are indicated by particles and adverbs clustered on the head. In addition, up to three auxiliary markers and directional markers may appear. These markers are derived from intransitive motion verbs that may appear as main verbs in other contexts.[2]

(1) *Q'anjob'al example*
Maxab' ek'elteq ix unin yet sq'inib'alil tu.

max-ab'	ek'-el-teq	ix	unin	y-et
COM-EV	pass-DIR-DIR	CL	child	E3S-when

s-q'inib'-al-il	tu.
E3S-early-ABS-ABS	DEM

"The child came out that morning (they say)."

3. A Finite-State Morphological Analyzer

Since so far no significant lexical resources exist for Q'anjob'al, the annotation of corpora and the compilation of a basic lexicon goes hand in hand. It is obvious that for this task, computational support can be very useful, generalizing from already annotated text to new occurrences

[1] http://www.sil.org/computing/shoebox/

[2] Note that the apostrophe forms a unit with the preceding letter.
ABS=abstract, COM=completive, CL=classifier, DEM=demonstrative, E=ergative, EV=evidential, S=singular, 3=third person

of the same word forms. With the morphological richness of the language, it turns out however that there are not many simple word-form based generalizations to be made, taking into account only the relatively small collection of text. Hence we applied a finite-state morphological analyzer, based on a very simple morphological grammar and a list of prefixes, stems, and suffixes which we compiled in a few hours by inspecting a number of frequent words. We also experimented with the unsupervised morpheme-inducing tool Linguistica[3] to generate candidates for affixes; however, with the direct access to native speaker intuitions it is not so clear whether such a tool provides a significant advantage.

The morphological analyzer was implemented using the Finite-State Automata Utilities for Prolog by van Noord.[4] The morphological grammar covers 96 affixes and 152 stems; it assigns a fine-grained morpheme label to each morphological element, and the Spanish translation to the stem entries. (2) shows the analysis for the aspectual verb 'lajwi' (stop), with the inchoative prefix 'ch'. The minimized finite-state transducer compiled from our morphological grammar consists of 2610 states and 9558 transitions.

(2) word form: chlajwi
morphol. analysis: ch @inc lajwi #vas =terminar

3.1. Discussion

As can be expected, the small morphological grammar we wrote is too limited to warrant a broad-coverage application. We did not implement very tight constraints on prefixation and suffixation, which leads to overgeneration but facilitated an initial exploratory analysis of word forms for the stems we covered. We considered expanding the small grammar to cover more than the initial set of stems, but we decided that our other experiments should have higher priority and discontinued this part of the project, at least for the current project phrase. While without doubt, a morphological analyzer would be extremely helpful, the development of a grammar of acceptable quality presupposes fairly advanced knowledge of finite-state technology and requires at least a few months of development effort. So, it is something that cannot be easily done in a typical language documentation project. If a high-level tool support could be provided for specifying finite-state morphological grammars, this might change the situation in the future.

4. Maximum Entropy Part-of-Speech Tagging

By tagging the Q'anjob'al text corpus for part-of-speech category information, we think we can increase its value considerably, both for linguistic and educational application. The tagged version of the corpus may also be the basis for additional NLP work on the corpus. The tagset was chosen to reflect the descriptive categories used in work on Q'anjob'al (the abbreviations follow their Spanish names). The table in figure 1 shows a part of the tagsets we assumed. The full set comprises 60 tags.

s	sustantivos (nouns)
snv	sustantivos no variables (nouns without variation in possession)
spr	nombres (proper nouns)
ssp	susntantivos siempre poseídos (always possessed nouns)
snp	sustantivos nunca poseídos (never possessed nouns)
srel	sustantivos relacionales (relational nouns)
scp	sustantivos compuestos (compound nouns)
v	verbos (verbs)
vif	verbos intransitivos flexionados (tensed intransitive verbs)
vin	verbos intransitivos infinitovos (infinitive intransitive verbs)
viax	verbos intransitivos auxiliares (auxiliar verb)
vidir	verbos intransitivos direccionales (directional intransitive verbs)
vtf	verbos transitivos flexionados (tensed transitive verbs)
vtn	verbos intransitivos inifinitivos (infinitive transitive verbs)
pos	posicionales (positional words)
adj	adjetivos (adjectives)
adv	adverbios (adverbs)
num	números (numbers)
cit	marcador de citas (quotative marker)
cln	clasificadores nominales (nominal clasifier)
clnu	clasificador numérico (numeral clasifier)
comp	complementizador (complementizer)
cor	marcador de cordinación (coordinate marker)
cond	marcador conditional (conditional marker)

Figure 1: Part of the tagset

4.1. Procedure

In accordance with our overall goal of exploring a breadth-oriented approach to resource creation, we did not intend to provide manual part-of-speech annotation of our entire corpus, but we experimented with machine learning techniques based on a relatively small manually labeled training corpus. Given this situation, it was natural to adopt a training technique that allows us to base learning on as many features as possible: the Maximum-Entropy approach. Crucially, Maximum-Entropy training does not build on an independence assumption for the features used, so one can provide a large selection of closely related features.

We followed the original Maximum-Entropy tagging set-up of (Ratnaparkhi, 1996) and used the YASMET tool by Franz Josef Och[5] for training the model parameters. This involved writing some additional Perl scripts that convert the training data into the required representation format, and a search algorithm for the application of the Maximum-Entropy model in actual tagging (on the test data etc.). We implemented the simple beam search algorithm of (Ratnaparkhi, 1996).[6]

[3] Goldsmith, http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/

[4] http://odur.let.rug.nl/˜vannoord/Fsa/

[5] http://www.isi.edu/˜och/YASMET.html

[6] (Bender, 2002) provides `MEtagger` a toolkit for part-of-speech tagging based on the YASMET implementation of generalized iterative scaling. It includes an implementation of a Viterbi search algorithm. Using `MEtagger`, it is possible to train a Maximum-Entropy tagger without having to do any program-

	Features	*Standard application*		*With confidence filtering (probability for*			
	used in	Accuracy	Unseen	*second-best tag < 30% of prob. for best)*			
	training		words	Words skipped	Precision	Recall	F_1-score
Tagger A	(3a-d)	**60.0%**	27.4%	23.3%	**73.0%**	55.3%	62.9%
Tagger B	(3a-l)	**80.2%**	27.4%	15.4%	**85.8%**	71.6%	78.0%

Figure 2: Comparison of results for two Maximum Entropy Taggers

We manually annotated about 4,100 words as training data and defined the following types of features as the basis for learning (the features described as "whether ..." are binary features):

(3) Classification task: assigning a part-of-speech category to a word w_i in a given text in Q'anjob'al.

Types of features for Maximum-Entropy Tagging:

a. the word w_i

b. the preceding words w_{i-1}, w_{i-2}

c. the subsequent words w_{i+1}, w_{i+2}

d. the tags assigned to the previous tags tag(w_{i-1}), tag(w_{i-2})

e. whether w_i is capitalized

f. whether w_i contains a digit

g. the prefixes of w_i (the 1, 2, 3, and 4 initial letters)

h. the suffixes of w_i (the 1, 2, 3, and 4 final letters)

i. word length of w_i (using features for various length levels)

j. the `mkcls` cluster to which w_i was assigned in unsupervised clustering

k. whether w_i is included in a wordlist for Q'anjob'al

l. whether w_i is included in a wordlist for Spanish

Features (3a-d) are the classical contextual distribution features. (3e-i) add features sensitive to the word-internal shape, as is standard in Maximum-Entropy tagging; in an experiment like ours, which is based on a small training set and for which no dictionary lookup of categories supports the tagger, such features taking up morphological generalizations are particular important.

The features in (3j-l) are special features exploiting some additional information that we could come by. For (3j), we used the `mkcls` tool by (Och, 1999) for unsupervised classification of the words in all our Q'anjob'al texts into 50 classes or clusters. This is a way of exploiting some of the information from the texts for which we could not provide manual part-of-speech information. For (3k), a wordlist of about 1,400 words of Q'anjob'al that was available to us is used; among other things, the list includes a fairly systematic list of body parts, family relations, plant names, and common adjectives. Although plain membership of the word w_i to be tagged in this list is not highly informative, this feature may nevertheless be helpful, e.g., for generalizing from the word for *hot* to the word for *cold*. It is also a good indication for native roots of Q'anjob'al. In order to be able to detect borrowings from Spanish we also provide a Spanish wordlist feature in (3l).

4.2. Results

We ran 10-fold cross-validation experiments, i.e., we split our training data in 10 parts and executed 10 subexperiments in each of which we reserved one part as the evaluation data and used the remaining 9 parts for training. In order to see what effect the use of morphological features and our special features had, we first trained the tagger using only the distributional features (3a-d) and then compared it to a tagger trained using the full set of features (3a-l).

We evaluated the accuracy of part-of-speech tagging, i.e., the proportion of word forms in the unseen evaluation data that were assigned the correct tag (i.e., the same tag as in the manual gold standard annotation). In addition we ran a different evaluation procedure, which essentially skipped word forms for which the trained tagger did not have very high confidence in its decision. We skipped word forms for which the probability for the second-most likely tag was 30% or more of the probability for the most likely tag. In our score tables we show the proportion of forms that are skipped by the procedure, and the precision and recall values (and F_1-score)[7] for the cases where the tagger did make a decision. We think that there are applications of the part-of-speech tagger for which high precision is more important than high recall (e.g., using the output of the tagger in order to provide a larger training set for retraining in a bootstrapping cycle). It is also conceivable to mark higher-confidence output of the tagger in a special way, so the reliability is transparent to the user.

The table in figure 2 summarizes the results for our first training experiment. Note that a high proportion (27.4%) of the word forms occurring in the evaluation set were entirely new to the tagger and we could not provide a look-up list of possible tags for these words (which makes the task very hard for a tagger that cannot use word-internal features).

With the full feature set, accuracy could be improved significantly. The number of word forms skipped for low confidence thus also decreases from 23.3% to 15.4%. But still, if precision is most important, the skip mode can be very useful, leading to 85.8% precision.

4.3. Discussion

With the availability of tools like YASMET, training of a part-of-speech tagger for a new language on a small train-

ming. (We had already implemented our own programs when we became aware of the `MEtagger` software; we continued using our programs for this study since this gave us more flexibility for experimentation, but `MEtagger` seems very appropriate for similar projects.)

[7] Precision = # tags correctly assigned / # tags assigned in total; Recall = # tags correctly assigned / # words in the evaluation set; F_1-score = 2 x Recall x Precision / (Recall + Precision)

ing set can be prepared and executed within a few weeks. Currently the process still requires some expertise in the underlying technology and at least some script writing, but it is conceivable that the necessary steps could be encapsulated in a higher-level toolkit.

Part-of-speech tag information is a highly useful annotation to provide for language documentation. Once the tagger has been trained, it can be applied to any new texts acquired for a consistent annotation. Although the accuracy we could reach with Maximum-Entropy training on a small data set is not as high as the rates familiar from taggers for the "big languages", we think that it provides a reasonable basis for accessing the language data in a much more systematic way than simple text search. By correcting the tagger output to produce a larger set of data, one may also bootstrap a tagger with higher accuracy.

5. N-Gram Language Modeling

The third technique from NLP research that we experimented with on our corpus of Q'anjob'al was N-gram language modeling. Here the idea is to train a statistical model that predicts how likely it is to find a particular word form as the continuation of the text after a number of given words. (Typically one only considers the two words preceding the word form in question.) Language models are widely applied in NLP, for example in speech recognition and machine translation, and there are toolkits available that make the creation of a language model, given a plain text corpus (without any further annotation), very easy.

A potential issue for language modeling on texts from documentary projects is that the amount of available training data may be too small to get a very reliable performance. Also, it is not immediately clear what a useful application in this context is. But since it is so easy to train a language model, one may at least use it as a tool for exploring certain aspects of the data.

We propose one specific use of a language model that turned out useful: detecting potential typos or orthographic inconsistencies (i.e., a very simple spell checker).

We used the CMU-Cambridge Statistical Language Modeling Toolkit[8] for training trigram language models on our full corpus (minus a short evaluation section). We preprocessed the corpus by a simple tokenization step, lowercasing all word forms. When we trained the language model, all word forms occuring only a single time in the training data were treated as the special "unknown" symbol ⟨UNK⟩. We used the standard Good-Turing discounting method. The perplexity of the language model with respect to the training data was 19.75 (entropy: 4.3 bits); the perplexity wrt. the held-out evaluation data was 337.32 (entropy: 8.4 bits; only 20% of the trigrams were known, for 40% back-off to bigrams could be used, in 40% of the cases, back-off based on unigrams was used).

In order to implement a very simple "spell checker", we wrote a Perl program that applies the language model to an input string and performs a special operation for unknown words: for these words, it checks whether any of the words in the vocabulary is (i) similar to the unknown word found, and (ii) more likely to appear in the given context. For measuring similarity, we use the Levenshtein distance (or edit distance).[9]

In our experiment, our "spell checker" reported only a small number of inconsistencies, 25% of which pointed to a real problem (i.e., there were real inconsistencies in the transcription of the text). This may be an acceptable proportion if the checking is applied during resource development, but presumably the text contained a larger number of inconsistencies that were not noticed by our tool.

5.1. Discussion

Training of a language model for a given corpus (in text file format) involves the least technical sophistication of the three experiments we conducted. The resulting model (and already the N-gram frequency lists) can be useful for an exploratory analysis of the data; however, we were not able to use the language model directly for the most important steps in corpus development, besides a small spell checking tool, which is presumably not critical enough.

6. Conclusion

We reported on experiments in the application of standard techniques from NLP in the context of a language documentary project. Specifically, we looked at three techniques: Finite-state Morphological Analysis (section 3.), Maximum Entropy Part-of-Speech Tagging (section 4.), and N-Gram Language Modeling (section 5.). In each case we made use of existing, freely available toolkits, and we had to do some limited additional programming. We were mainly interested in techniques that did not involve time-intensive development of background resources such as grammars.

We reached the most far-reaching results with training of a Maximum Entropy Part-of-Speech Tagger, based on a rich set of features.

7. References

Bender, Oliver, 2002. Untersuchung zur Tagging-Aufgabenstellung in der Sprachverarbeitung. Diploma thesis, Lehrstuhl für Informatik VI, RWTH Aachen, Aachen.

Och, Franz Josef, 1999. An efficient method for determining bilingual word classes. In *Proceedings of the Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics, EACL'99*. Bergen, Norway.

Ratnaparkhi, Adwait, 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*. University of Pennsylvania.

Richards, Michael, 2003. *Atlas Lingüístico de Guatemala*. Guatemala: Universidad Rafael Landívar.

[8] http://mi.eng.cam.ac.uk/˜prc14/toolkit.html

[9] We apply the Perl implementation by Eli Bendersky: http://www.merriampark.com/ldperl.htm