

Recognizing Proper Names in UR III Texts through Supervised Learning

Yudong Liu, James Hearne and Bryan Conrad

Computer Science Department
Western Washington University
Bellingham, Washington 98226

{yudong.liu@-james.hearne@-bconrad@students.}www.edu

Abstract

This paper reports on an ongoing effort to provide computational linguistic support to scholars making use of the writings from the Third Dynasty of Ur, especially those trying to link reports of financial transactions together for the purpose of social networking. The computational experiments presented are especially addressed to the problem of identifying proper names for the ultimate purpose of reconstructing a social network of UR III society. We describe the application of established supervised learning algorithms, compare its results to previous work using unsupervised methods and propose future work based upon these comparative results.

Introduction

The Third Dynasty of Ur was a 21st to 20th century BC Sumerian ruling dynasty based in the city of Ur. It was a period of unprecedented economic expansion and was marked by advances in record-keeping that has left a vast archaeological record of commercial and social relations. This record arouses the possibility of investigations into social and economic history not possible for most ancient societies. The vast majority of clay tablets from the Third Dynasty of Ur, also known the Ur III period, record financial transactions, such as records of cattle deliveries, receipt of metals, repayment of loans and so forth. To press these records into the service of social history requires linking tablets into larger mosaics of summative information. Some linkings require only keyword searching, though it should be noted in passing that the two main repositories provide only primitive keyword searching, not enabling, for example, simple Boolean queries. In any case, the present effort goes far beyond what might be hoped for in the way of keyword searching.

One of ways of linking tablets to support of an aggregate understanding of their role in an ancient social network is to re-identify the same actor in different tablets. Importantly, in addition to the provider and recipient of transference, tablets consistently enumerate lists of witnesses. This fact makes the tablets an invaluable resource for the social history of the time since they record, implicitly, on each tablet, lists of persons who knew one another and who enjoyed professional relations with one another. The recovery of personal

names on the tablets suggests the possibility of reconstructing social networks of actors in the mercantile class and also, given the overlap, their social network connections to royalty. Use of the tablets for such purposes is impeded in three ways. First, they are by and large written in Sumerian, a language already going out of use as the language of the streets at the time the tablets were being written and which, even after nearly two centuries of scholarly investigation, is ill-understood. A casual exploration conducted by us showed that roughly half of the lexical items found in the UR III Corpus are *hapax-legomena*, words having but a single attestation, a fact that greatly impedes decipherment. The problem is further compounded by the fact that the tablets come down to us in many cases damaged by time and the circumstances of their recovery which was, in many cases, looting. Second, although there are now scholars able to make sense of tablets relating to merchant records, the corpus is of a size too large for even a community of scholars to master. One source estimates over 90,000 published tablets and tens of thousands yet unpublished (Widell 2008); further, new tablets are still being uncovered, even though the region they are found in, modern Iraq, is now in its second decade of open warfare. Third, the project of linking tablets through proper names presupposes sufficient density in the name space so that their reidentification is likely.

Background

The research on how to make use of natural language processing (NLP) methods to process an ancient language like Sumerian is not voluminous. Easily available contributions are confined to (Tablan, 2006) and (Jaworski, 2008). Other applications to NLP have been published but apparently have been withdrawn, especially since this language is no longer in use. Unlike Greek or Chinese, which, though having long histories, are still in use, it is hard to obtain the information or learn about the language. In this section, we introduce the Sumerian databases and the previous study on Sumerian text.

Sumerian Databases

There are two main Sumerian tablet databases, the Cuneiform Digital Library Initiative (CDLI, <http://cdli.ucla.edu/>) and the Database of Neo-Sumerian Texts (BDTNS, <http://bdtns.filol.csic.ed/>). CDLI is a

project that concentrates on electronic documentation of ancient cuneiform, consisting of cuneiform texts, images, transliterations and glossaries. It is managed at UCLA, USA and the Max Planck Institution for the History of Science, Berlin. BDTNS is a database that manages more than 95,300 administrative Sumerian cuneiform tablets during the Neo-Sumerian period.

Previous Work on Sumerian Texts

Research on the application of machine learning and NLP techniques to ancient languages is not abundant. By and large, the application of computer technology has been limited to electronic publishing and string searching. Such efforts applied to Sumerian are even more sparse. (Tablan et al. 2006) described the creation of a tool for Sumerian linguistic analysis and corpus search applied to Sumerian literature. This work focused on the application of a morphological model for noun and verb recognition, developed by Sumerologists. However, it did not extend beyond simple word and morpheme recognition. (Jaworski 2008) developed an approach to extraction of information from the same economic documents of concern to us by beginning with an ontology of the world presupposed by this corpus and, in conjunction with syntax and semantic analysis. The claim of this work is that it supports research into the Sumerian language, officials participating in the activities the tablets record, and into document classification. Our previous work includes (Brewer et al. 2014) and (Luo et al. 2015). Both are addressed to the recognition of proper names in Sumerian text. (Brewer et al. 2014) uses a naive edit-distance and pattern-matching algorithm to attempt to identify personal names. (Luo et al. 2015) applies a DL-Cotrain based unsupervised learning method with 3 seed rules. This unsupervised method achieves a high recall (92.5%) and a low precision (56.0%). Some comparisons among these work are described in the later section.

Data of CDLI

In this project, we still use the CDLI repository as we did in our previous work for its restriction to the ASCII character set is more convenient than the UTF-8 encoding used by BDTNS, as well as for comparison purpose. Of this repository we use the 53,146 tablets having lemmata.

In the following, we will give more detailed introduction of the CDLI data.

CDLI: Words and Signs

Although the texts we are investigating were originally written in cuneiform script, scholars have traditionally worked with transliterations using the English alphabet. Homophony, which is very common in cuneiform, is handled by numerical subscripts, i.e., “*gu*,” “*gu*₂” and “*gu*₃” refer to distinct signs with the same sound value. Sumerologists have developed a system of in-text annotations important to the application of NLP techniques.

Figure 1 shows the tablet with an id of P105955 from CDLI repository (<http://cdli.ucla.edu/>). The original

cuneiform script is on the left; the transliteration is in the middle and the modern English translation is on the right.

Sumerian written in cuneiform script does not have the concept of upper- or lowercase, and as a result, in monolingual contexts, the typical step of case normalization is not necessary as all text is rendered in lowercase in the transliteration format used by CDLI. However, signs rendered in uppercase occur frequently and denote a number of special circumstances, most commonly, that the phonetic equivalent of the sign is unknown. With respect to our system, the presence of uppercase signs is rarely an issue as long as the spelling and casing is consistently attested in the corpus. This consistency is provided by the lemmatization required of Sumerologists in order to submit initial transliterations (Foxvog 2014).

Royal epithets notwithstanding, Sumerian personal names are exclusively comprised of a single word, almost always consisting of at least two signs. In cases where disambiguation is required, a patronymic may added (for example, *szu-esz4-tar2 dumu zu-zu*, “Su-Estar, son of Zuzu”). This disambiguation is frequent in practice due to the relatively shallow pool of personal names used (Limet 1960).

Lemmata

62.1% of the tablets (53,146 tablets) in the CDLI corpus are accompanied by annotations, dubbed “lemmata” which provide, stemification, translation and categorization of linguistic elements within a line. Thus, the example given in Figure 2 gives a line of text followed by its lemmatization, indicated with a line-initial “#.”

In this example, the lemmatization indicates that the word “GAN2” derives from the Sumerian stem “iku” and is understood to mean “unit.” Similarly, “ur-{gesz}-gigir” is a personal name; “nu-banda3” with the Sumerian stem “nubanda” means “overseer”, a common profession in Sumerian society; the word “gu4” with the Sumerian stem “gud” means “ox.” Sometimes, when the function of a work is uncertain, the lemmatization offers more than one category. For example, the lemma “GN|FN” indicates that the corresponding can be either a geographical name or the name of a field (analogous the “Potter’s Field.”).

Damaged Tablets

In CDLI corpus, some transliterations have “[” and “]” attaching to a sign. It indicates the sign is damaged. More specifically, “[” indicates that the following sign is damaged on the left edge, whereas “]” indicates the following sign is damaged on the right edge (Sahala 2012). “[x]” indicates that there is a sign that cannot be read due to the damage on both edges and “[...]” indicates that there are several unreadable signs.

Supervised Proper Name Recognizer

An important task in supervised learning is to propose a set of useful features. Because there is no previous work as such, we proposed a set of features that capture either the context or the spelling of the word of interest. We reported the system performance of a variety of supervised learning



Cuneiform Tablet	Transliteration	English Translation
	&P105955 = BIN 03, 149	(unique tablet identification)
obverse	@tablet	
	@obverse	(front of tablet)
	1. 1(disz) sila4	1 lamb
	2. ki ab-ba-sa6-ga-ta	from Ab-ba-sa-ga (the seller)
	3. ur-{d}szul-pa-e3	Ur-Šul-pa-e (the buyer)
	@reverse	(back of tablet)
reverse	1. i3-dab5	received
	2. iti u5-bi2-gu7	month: 3
	3. mu hu-uh2-nu-ri{ki} ba-hul	year: Huhhuri was destroyed
	@seal	(tablet sealed by)
	1. ur-{d}szul-pa-e3	Ur-Šul-pa-e
	2. dub-sar	the scribe
	3. dumu da-a-da kuruszda	son of Da-a-da the fattener

Figure 1: Tablet with ID of P105955 from CDLI.

4. GAN2 ur-{gesz}gigir nu-banda3 gu4
#lem: iku[unit]; PN; nubanda[overseer]; gud[ox]

Figure 2: A lemmatized transliteration.

strategies. These learning strategies did not show much difference in performance. We also compared our supervised learning result with the unsupervised learning result reported in (Luo et al. 2015), and gave some discussion. In the following, we will describe the system in more details.

Data

To utilize the pre-knowledge from the language experts and (Weibull 2004), we apply a tag set of 13 tags to pre-annotate the corpus. The 13 tags in the tag set {"GN", "FN", "TN", "WN", "MN", "n", "TITLE", "UNIT", "GOODS", "OCCUPATION", "YEAR", "MONTH", "DAY"} represent geographical names, field names, temple names, watercourse names, month names, numbers, title names, unit names, trade goods names, occupation names and indicators for year, month and day, respectively. In the following, we call such tags "part-of-speech tags".

For the purposes of this work, the Ur III corpus was divided into a number of subsets. The training set, used to train all our machine learning models, consisted of 80% of the portion of the corpus for which there were reliable part-of-speech tags. All lines which contained any damage to the words were then removed, so as to be as certain as possible that the features and labels the models were trained with were accurate. As a result, there are 1,788,627 training examples in total of which 9.5% are positive examples and the rest are negative examples.

The remaining 20% of the labelled corpus was used for two different test sets. Test set 1 consisted entirely of lines containing no damage. In total we have 443,264 examples in the test set 1. Test set 2 contains all of set 1, along with some lines (about 10,000) with some damage. However,

only words with damage the transliterators judged was minor enough that they could still make a reasonable assertion about what the word was (and it's part of speech) were allowed. There are 447,995 examples in the test set 2, which indicate the 4,731 examples that are not in test set 1 but in test set 2 are words that contain damaged signs.

Features

For each word in the corpus, the presence or absence of 36 features was recorded. The majority of the features are either context-related features such as the word or its part-of-speech tag to the left or the right of the current word, or the spelling features such as if the current word contains a certain sign or a certain part-of-speech tag (such as "profession"). Other features include the positional information about the current word in the current line, the repetitive occurrences of certain signs in the current word and if the current word is the only word of the line. The contextual features and spelling features are mostly overlapped with the contextual rules and spelling rules learnt from the DL-Cotrain method in (Luo et al. 2015). Table 1 enumerates these features.

Supervised Learning

Seven machine learning algorithms were chosen for this work, all of which implemented by sci-kit learn (Pedregosa et al. 2011). Each implementation has a number of options for customizing its performance, such as learning rate and number of training iterations. The values for these parameters were chosen by first selecting a list of reasonable values for each option, then for each combination of values a model was trained, and the model giving the best accuracy was kept. The algorithms are:

1. Decision Tree Classifier: A single decision tree constructed with a version of the Classification and Regression Trees (CART) algorithm, using information gain (Kullback-Leibler divergence) to determine the optimal splits.

word index in line. 0-indexed.
left context. None if this is the first word in the line.
right context. None if this is the last word in the line.
line context. All words in line, space-delimited.
if word is alone on line.
if left context is **dumu** (“child (of)”).
%This may suggest a personal name in a patronymic.
if right context is **dumu** (“child (of)”).
if line context is **ki** word.
%This may suggest a seller in a transaction.
if line context is **igi** word.
%This may suggest a witness to a transaction.
if line context is **igi** word-**szē3**.
%This may suggest a witness to a transaction.
if line context is the Personnenkeil **1(disz)** word.
%This may suggest a list of named individuals.
if line context is **kiszib3** word.
*%This may suggest the individual responsible for sealing
%a tablet.*
if line context is **giri3** word.
if first sign in word is repeated.
%Sumerian names tend to favor repeated syllables.
last sign in word is repeated.
if any sign in word is repeated.
if word is a common profession.
if word contains a profession.
if left context is a profession.
if left context contains a profession.
if right context is a profession.
if right context contains a profession.
if word starts with **ur**.
if word starts with **lu2-**.
if word ends with **-mu**.
if word contains {**d**}.
%short for “dingir” (“deity”), the divine determinative.
if word contains {**ki**} “place”.
if word contains any determinative.
if word’s transliteration contains the letter **q**.
if word contains **lugal** (“king; large”).
if word contains a number.
if word followed by **sag**.
if word followed by **zarin**.
if word followed by a numeric classifier.
if line context begins with **iti** (“month”).
if line context begins with **mu** (“year”).

Table 1: 36 features applied in training data and test data

2. Gradient Boosting Classifier: Combines 500 weak regression trees using a logistic regression loss function into a stronger classifier. This is done in sequence, with each trees contribution to the overall result being smaller than the size of the previous one by the learning rate, which is 10%.
3. Logistic Regression: Training a logistic regression model consists of using Maximum Likelihood Estimation to find a series of weights such that the logistic function, when

given the weighted sum of the input features, gives results close to 1 when the true class is “positive” or “success” (In the case of this work, a personal name) and close to 0 otherwise.

4. Naive Bayes: Makes use of Bayes Theorem to train a simple classification model. Starts with the observed data (including the distribution of names/non-names in the training set), and calculates the most probable class for each word. Despite the assumption that the features are all conditionally independent given the true class, Naive Bayes has enjoyed some success as a classification tool.
5. Stochastic Gradient Descent (SGD): Does Stochastic Gradient Descent (i.e. updates the gradient of the loss with each instance examined, rather than averaging it across all instances in the training set) on a logistic regression model.
6. Linear Support Vector Machine: Support Vector Machines try to discover a hyperplane across the set of features that divides the data points in the “personal name” class from the others or, if it cannot, to find a hyperplane that misclassifies as few points as possible (called a soft margin).
7. Random Forest Classifier: A collection of 500 decision trees, trained on different subsets of the data and combined to form a stronger classifier. Each tree contributes its estimation of the probabilities of the different output classes to the final result.

Results

For the most part, the results obtained from the different algorithms were quite similar, though there were some interesting patterns.

Method	Precision	Recall	Fscore
Decision Tree	0.898	0.475	0.621
Gradient Boosting	0.872	0.660	0.752
Logistic Regression	0.858	0.662	0.748
Naive Bayes	0.831	0.611	0.704
Gradient Descent (SGD)	0.864	0.653	0.744
SVM	0.856	0.655	0.742
Random Forest	0.932	0.458	0.614

Table 2: Test Set 1 - data with no damaged words

Method	Precision	Recall	Fscore
Decision Tree	0.899	0.474	0.621
Gradient Boosting	0.873	0.658	0.750
Logistic Regression	0.859	0.660	0.746
Naive Bayes	0.832	0.609	0.703
Gradient Descent (SGD)	0.864	0.650	0.742
SVM	0.857	0.653	0.741
Random Forest	0.933	0.458	0.614

Table 3: Test Set 2 - data with damaged words

Each classifier produced very similar results on both data sets. This indicates that the damaged signs that can be annotated in the lemmata are also easy for the machine to recognize through the context or the spelling components. It would be interesting to see how the system works on the damaged signs that are not initially marked in the lemmata. However the evaluation on such data set would be more challenging given that it would involve more human expertise.

The Decision Tree classifier and the Random Forest classifier (which was constructed from multiple decision trees) both traded low recall for a higher precision to a greater extent than the others, in particular the Random Forest, which had both the highest precision and lowest recall of all the algorithms. Of the others, the Gradient Boosting classifier, Logistic Regression, Stochastic Gradient Descent, and the Support Vector Machine all produced very similar results. The Naive Bayes classifier, on the other hand, does not seem to be the best model for this task, at least as used here. It had the lowest precision, at 83%, and the lowest recall after the models involving decision trees, at 61%.

The similarity between the results of different methods raised suspicions that they were learning very similar models, and that many words labeled correctly as personal names were identified by all or most of the algorithms. To test this, a system was set up in which each algorithm's label for a word was interpreted as a 'vote' that that class be assigned to that word, and each word was classified according to the label that received the most votes across all methods. This conjecture seems to be confirmed, although more investigation is still under way; the resulting system achieved a recall of 65% and precision of 86% on both test sets.

Comparison with the Unsupervised Method

(Luo et al. 2015) reported a DL-CoTrain based unsupervised Proper Name Recognizer for CDLI data set. With 3 seeds rules and 150 iterations, this system generated over 2,000 decision rules, and achieves a high recall (92.5%) and a low precision (56.0%), which means it produces a lot of false positives. This indicates that a subset of these over 2,000 rules are accurate enough that it predicts most of the true names. However it also generates a lot of rules which are not accurate enough that 44% of the names it predicts are not labelled as names by the annotators.

In the supervised setting, the system exhibits opposite behaviors. It achieves a low recall (around 65%) but a high precision (around 86%), meaning that the supervised system can predict a name with a high accuracy but is not competent enough to identify much more than a simple majority of the names. This may be due to the fact that the features used are very good ones when they apply but their number and variety are insufficient to provide full coverage. Thus, we need a larger feature set.

Some of the 36 features in the current supervised setting are selected from the high confident contextual rules and spelling rules generated by the DL-Cotrain method. One possible way to improve the current supervised learning result is to include more features from the decision list of DL-Cotrain.

Conclusion and Future Work

In this paper we reported our ongoing work on a supervised proper name recognizer of the Sumerian corpus of CDLI database.

As the first work of applying supervised method on Sumerian proper name recognition, our major contribution is two-fold. One is in feature selection. With the result from our previous work on unsupervised learning on this task, and with the help from a domain expert in our home university, we proposed a set of 36 features for both training and testing. The experiments have shown that different machine learning methods, when exploiting these features, can perform with high precision, which indicates the features are accurate enough to predict true names. Secondly, our work shows that the decision tree based method is not the best option in this setting, which indicates that the combinations of features are more predictive of the class label in this application. However the decision tree do not provide such information. The relative poor performance of the Naive Bayes method may be due to the fact that some of the features here are more correlated with each other. Other machine learning methods gave very similar result and performance. In comparison with the previous work on unsupervised Sumerian proper name recognition (Luo et al. 2015), it suggested that the current feature set is accurate but not large enough. How to obtain a richer feature set, compare to the adoption of a machine learning strategy, is more important for this task.

For the future work, we would like to do a more in-depth comparison between the features we proposed here and the rules resulting from the unsupervised method. We believe that it would help to further incorporate good features. It would also be interesting to do analysis of the relevance of the features. More feature engineering and analysis would tell us what types of features are more salient and worth exploring. Since our work has drawn more attention from the Assyriologists across the world, we have been provided a richer set of features on this task. In addition, our experimental results showed that some methods tend to favor precision, while others recalls. We need to do more exploration on how the training process and parameter optimization affected the results, and see if the process has been sufficiently optimized.

Acknowledgments

The authors wish to thank Steven Garfinkle from Department of History of Western Washington University, and Mr. Clinton Burkhart for their assistance and valuable input as a language expert in this work.

References

- Brewer, F.; Burkhart, C.; Houg, J.; Luo, L.; Riley, D.; Toner, Brandon. Liu, Y.; and Hearne, J. 2014. A preliminary study into named entity recognition in cuneiform tablets. In *The third Pacific Northwest Regional Natural Language Processing Workshop*, 1–3.
- Foxvog, D. 2014. An introduction to sumerian grammar.

- Jaworski, W. 2008. Contents modeling of neo-sumerian ur iii economic text corpus. In *Proceedings of the 22nd International Conference on Computational Linguistics*, 369–376.
- Limet, H. 1960. *L'Anthroponymie sumerienne dans les documents de la 3e dynastie d'Ur*. Paris: Soci  t   d'  dition Les Belles Lettres.
- Luo, L.; Liu, Y.; Hearne, J.; and Burkhart, C. 2015. Unsupervised sumerian personal name recognition. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015, Hollywood, Florida. May 18-20, 2015.*, 193–198.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.
- Sahala, A. 2012. Notation in sumerian transliteration. Technical report, University of Helsinki.
- Tablan, V.; Peters, W.; Maynard, D.; and Cunningham, H. 2006. Creating tools for morphological analysis of sumerian. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, 1762–1765.
- Weibull, N. 2004. A historical survey of number systems. Technical report, Chalmers University of Technology.
- Widell, M. 2008. The ur iii metal loans from ur. In Garfinkle, S., and Cale Johnson, J., eds., *The Growth of an Early State in Mesopotamia: Studies in Ur III Administration*. Madrid: Consejo Superior de Investigaciones Cient  ficas. 207–223.