Using LPath Queries to Annotate Corpora: A Case Study of Elamite and Sumerian

Eric J. M. Smith

University of Toronto

1. Introduction

For many languages, the resources to properly annotate a corpus are simply not available, so the linguist has only a corpus of raw text to work with. In such cases, the linguist has traditionally had to expend considerable extra effort, manually combing through the raw text of the corpus to find the forms which are of interest. Even when the corpus is available in electronic form this can be a laborious process, one which is at best a diversion from the actual task of linguistic analysis.

When, as is the case with cuneiform texts, the writing system does not clearly represent the spoken form, the linguist faces additional obstacles. Even with an electronic corpus and the best available search tools, locating particular morphemes using only orthographic strings can be a tedious and repetitive task.

Since the types of queries necessary to locate morphemes do tend to be repetitive, it is helpful to wrap useful queries in a form which allows them to be easily reused. In addition, once the set of queries necessary to identify a particular morpheme has been established, that query can be stored and used as the basis for further queries.

Over time, a library of such queries can built up, with lower-level queries for individual morphemes being used to construct higher-level queries that identify syntactic structures of interest. Since each of the stored queries corresponds to a linguistic element, this library of stored queries effectively serves as an annotated representation of the corpus, one which was created without actual manual annotation.

1.1 Motivation

The author's primary interest is not in corpus linguistics, but rather in the morphosyntax of agreement. The underlying motivation for the work described here was research into the agreement morphology of Elamite and Sumerian. Both these languages show rather exotic agreement behaviour, and understanding that behaviour will increase our understanding of how agreement works cross-linguistically.

The difficulty is that there is no easy way to get at the relevant agreement morphology. Corpora for these languages (where they exist at all) consist of transliterations of the original cuneiform texts, with no morphological annotation. To make matters

worse, the orthographic systems of Elamite and Sumerian represent morphemes in a rather haphazard fashion, so identifying a morpheme from a string of graphemes is not a trivial task.

The task is further complicated by the type of morphosyntax being studied. By its nature, searching for agreement morphology requires the ability to search for discontinuous elements within a corpus. This makes the search problem significantly more difficult. Not only are the relevant morphemes obscured by the orthographic system, but they may be separated by an unknown amount of intervening material.

1.2 Methodology

The original strategy was to start with a powerful, flexible query language and use that as a basis for further development. The most promising candidate for such a query language was LPath Bird et al. 2005, Bird et al. 2006. Initial work indicated that using raw LPath queries to extract the desired agreement morphology proved to be unmanageably complex, largely due to the peculiarities of the writing system. To make the task more manageable, a new layer of reusable *query objects* was created, to encapsulate complex LPath queries into a more manageable form.

These query objects closely reflect the language's morphology. The end result is that the query objects fill in for the morphological annotation which is missing from the underlying corpus.

2. Corpora

The first decision was which corpora to use for the study. In the case of Elamite, there are no publicly available corpora of any significance. In the case of Sumerian, there are a number of possible corpora, and the choice of corpus hinged on the amount of metadata provided.

The approach being described could in theory be bootstrapped on top of a corpus which consisted solely of transliterated text. However, to keep the task manageable it helps to have at least a minimal amount of linguistic annotation. In particular, having a corpus which is already tagged for part-of-speech makes the queries considerably simpler. This was a major consideration when selecting the corpus, since the task of part-of-speech tagging would significantly increase the amount of work.

2.1 Elamite

The Electronic Corpus of Elamite Texts (ECET) was developed by the author Smith 2004 to store information about Elamite lexical items for research into the language's phonology Smith 2007. This was extended for syntactic research Smith 2006 to include a significant body of Elamite-language texts. This corpus encodes

both orthographic and morphological information, as well as translations of most texts.

The current ECET corpus consists of 221 texts, comprising approximately 20-000 words. Texts date from the Treaty of Narām-Sîn (ca. 2250 BCE) to the reign of Artaxerxes II (ca. 360 BCE). Due to the nature of the texts which have been recovered from Elamite archæological sites, the bulk of them are royal inscriptions, primarily transcribed from König 1965. There are also a number of small texts assembled from the Mémoires de la Délégation en Perse and other sources Scheil 1907, Scheil 1911, Scheil 1917, Paper 1954, Lambert 1974, Grillot-Susini et al. 1993, Vallat 1996. Although this corpus is small by the standards of corpus linguistics, it does represent a significant portion of all extant Elamite-language texts.

The majority of the corpus has been lemmatised and tagged for part-of-speech. Although much of this process was automated by performing lookups in the electronic version of the *Elamisches Wörterbuch* (Hinz and Koch 1987), the task of manually annotating all the ambiguous word forms has proven to be quite laborious.

Work is underway to add more bulk to the corpus, using texts from other Elamite-language research projects: the Italian-Iranian DARIOSH project (Achaemenid royal inscriptions) and the University of Chicago's Persepolis Fortification Tablets (Achaemenid economic tablets). Since the ECET corpus is still under construction, the remainder of this paper will be devoted to Sumerian.

2.2 Sumerian

For Sumerian there are a number of available electronic corpora. By far the largest collection of texts is the Cuneiform Digital Library Initative (CDLI) from UCLA and the Max Planck Institute (Englund and Damerow 2000). It has a broad range of texts from all periods, but the focus of the project is archæological rather than linguistic. Consequently, the entry for each text contains catalogue information, provenance, and images, but the texts themselves are only provided in transliteration with no translation or morphological markup. This is also true of a number of other smaller corpora associated with the CDLI, such as the Digital Corpus of Cuneiform Lexical Texts (Veldhuis 2003) and the Database of Neo-Sumerian Texts (Molina 2002).

Of great interest was the Pennsylvania Parsed Corpus of Sumerian (Tinney and Karahashi 2003), which was conceived as a hand-parsed treebank in the mould of the English-language Penn Treebank. Such a corpus would have been close to ideal for the purposes of identifying the morphosyntax of agreement. Unfortunately, work on the corpus seems to have stopped, and the corpus has never been publicly released. Inquiries with the project's staff indicate that the corpus never got beyond the pilot stage.

2.2.1 ETCSL

In the end, Oxford's Electronic Text Corpus of Sumerian Literature was selected as being the easiest of the Sumerian corpora to work with. In addition to transliterations, the corpus provides English translations, and the Sumerian text has already been lemmatised and tagged for part-of-speech.

The ETCSL consists of 394 texts from genres which Sumerologists classify as "literary": mythological epics, royal praise poems, literary letters, laws, hymns, cult songs, and proverbs. The corpus totals approximately 170 000 words of text. While 170 000 words is not a large corpus by the standards of corpus linguistics, for Sumerian it is quite substantial.

The majority of the texts date from a fairly narrow period (ca. 2200--1600 BCE), so the corpus is quite cohesive. Where variants exist they have been edited by the team at Oxford into a standardised form.

The XML source files for the corpus were made available by Jarle Ebeling and his colleagues. The corpus is organised as shown in (ETCSLStructure), with the top level being the `<text>`, which represents a self-contained document, possibly several hundred of lines long. Below the `<text>`, some of the documents are further subdivided using `<div1>` tags (used when there are lacunæ in the text) and `<lg>` tags (to group lines in certain genres, such as proverbs within a proverb collection). These intermediate groupings are not reliably present.

(1) Hierarchical structure within ETCSL
[Top-level] `<text>`
[Intermediate groupings] `<div1>`, `<lg>`
[Lines] `<l>`
[Words] `<w>`

The one grouping which is reliably present is the line, `<l>`. Unfortunately, in cuneiform texts there is no particular correlation between lines and sentence boundaries. The line is purely a scribal unit and may only incidentally correspond to a linguistic unit. The lack of phrase or sentence boundaries is a significant disadvantage for investigating syntactic questions, since the phenomena being explored are expected to be scoped to a single clause or sentence.

A typical word entry from the ETCSL is shown in (SampleWord). At first glance, the ETCSL provides a fair bit of morphological annotation. The *bound* attribute seemed particularly promising, since it promises a morpheme-by-morpheme breakdown of each word. Unfortunately, the *bound* attribute is only present on a handful of forms (ergative-case nouns for instance). Similarly, the *form-type* attribute is not as useful as it might be because it too is used for only a limited range of forms.

(2) A sample word entry from the ETCSL

```
<w form="nu-gi4-gi4" lemma="gi4" pos="V" label=
"to return" form-type="RR">nu-gi4-gi4</w>
```

[form] orthography
[lemma] standardised citation form/lexeme
[pos] part of speech
[type] further sub-grouping of *pos* (e.g. PN, DN)
[label] English gloss
[form-type] morphological information on word (e.g. reduplicated)
[bound] segmentational information (e.g. ergative-case suffix)

3. Queries

3.1 LPath Query Language

The query language being employed is LPath, developed by Steven Bird and his colleagues at the University of Pennsylvania. Bird's work with query languages started with the investigation of query languages for annotation graphs (Bird et al. 2000). In the past few years, he has turned to tree-structured data, and enhancements to a standard XML search language called XPath (Clark and De Rose 1999). The XPath language is intended for locating nodes within tree-structured XML documents, so it is a natural match for the task of locating elements within tree-structured linguistic data.

LPath does extend XPath somewhat by adding a variety of search operators which are useful for the kinds of searches done in linguistics. These are shown in (3).

(3) LPath operators added to XPath (Lai and Bird 2006)

- `->` (immediate-following) and
  `<-` (immediate-preceding)
- `=>` (immediate-following-sibling) and
  `<=` (immediate-preceding-sibling)
- `^` (left-edge alignment) and
  `$` (right-edge alignment)
- `{` and `}` (subtree-scoping)

A reference Python implementation of LPath is provided as part of the Natural Language Toolkit (NLTK) (Bird et al. 2007), an open-source collection of Python-language tools for computational linguists. Since Steven Bird is involved with both the NLTK project and with LPath, the NLTK is an appropriate place for LPath to be made publicly available.

Some sample LPath queries are shown in (4). The first one searches for a sentence, S, and that sentence must contain some entity (indicated by the underscore) which

has a *lex* attribute with the value of "saw". The second query is straightforward, locating nouns which follow a verb which is itself the child of a verb-phrase. The third query gives an example of the braces used to restrict the scope of a search to a subtree. The fourth through sixth demonstrate how the ^ and $ edge-alignment operators can be used to search for particular structural configurations.

(4) Sample LPath queries (Lai and Bird 2006)

1. `//S[//_[@lex=saw]]`
   A sentence containing the word 'saw'.
2. `//VP/V-->N`
   Nouns that follow a verb which is a child of a VP.
3. `//VP{/V-->N}`
   Within a verb phrase, nouns that follow a verb which is a child of the given verb phrase.
4. `//VP{/NP$}`
   Noun phrases which are the rightmost child of a VP.
5. `//VP{//NP$}`
   NPs which are rightmost descendants of a VP.
6. `//VP[{//^V->NP->PP$}]`
   Verb phrases composed of a verb, a noun phrase, and a prepositional phrase.

Although LPath is intended as an extension of XPath, this is not strictly true of the NLTK's LPath implementation. In particular, Xpath includes a large number of built-in utility functions for string operations, type conversions, and other operations. The LPath implementation lacks these functions, which is unfortunate since some of the basic functions (e.g. `substring`) would have been very useful in certain queries against the ETCSL corpus. Fortunately, the LPath implementation does include undocumented support for wild-card access using the SQL `like` operator, which provides a stand-in for some of the missing XPath string functions.

3.2 Adapting ETCSL for LPath

The ETCSL was made available by Oxford as a collection of XML files, but although LPath is based on an XML search language, the reference LPath implementation does not work with XML files, but rather with data stored in a relational database. In order to load the XML data into a MySQL database a small Python program had to be written.

The Python program required to load the database also provided an opportunity to massage the data somewhat, in order to make some of the more important pieces of information easier to access. In particular, prefixes, suffixes, and reduplication were identified and stored as separate attributes of each word. Early experimentation with the corpus suggested that it was useful to be able to refer to these items separately.

LPath queries are easier to write given the knowledge that a particular grapheme is in the prefix or the suffix.
For instance, given a complex verb form ⟨ ba-an-ci-gir$_5$-gir$_5$-e ⟩, the loader program can use the ETCSL's information that the lemma is *gir*$_5$ to extract the prefix *ba-an-ci-* and the suffix *-e* , and to recognise that the stem itself is reduplicated.
One significant piece of information which is missing from the ETCSL is information on sentence and clause boundaries. Although clause boundary information was unavailable, it was possible to determine paragraph boundaries, because the ETCSL indicates which lines of the transliteration correspond to a paragraph in the English translation. For many queries, a paragraph boundary serves as an acceptable proxy for a clause boundary.

3.3 Query Objects

In practice, the LPath queries that are needed to extract particular morphemes can become quite cumbersome. Consider the data shown in (Genitive), which shows only a few of the ways in which the genitive case suffix *-ak* might be written in a Sumerian text. This demonstrates the sort of mismatch which exists between Sumerian orthography and the language's morphology. Even if it were possible to write a very complicated LPath query which located all the possible orthographic forms for the genitive suffix, such a query would have to be used every time we wanted to search for a genitive-case noun, which would simply not be practical.

(5) Some orthographic realisations of the genitive case suffix *-ak*

- stem-final vowel assimilates to /a/ (e.g. 𒂵 ⟨ ĝa ⟩ after stems ending in ⟨ ĝu ⟩)
- 𒆷 ⟨ la ⟩ after stems ending in /l/.
- 𒈾 ⟨ na ⟩ after stems ending in /n/.
- 𒊏 ⟨ ra ⟩ after stems ending in /r/.
- sometimes written as 𒀀 ⟨ a ⟩
- only reflects the /k/ when before another suffix (e.g. 𒈗𒃲𒆷𒀀 ⟨ lugal-la-ke$_4$ ⟩ 'of the king-ERG')
- etc.

Recognising this problem, the approach was to incrementally build up a definition of a genitive-case noun using a series of queries. The queries in (BuildingQueries) correspond roughly to the orthographic forms shown in (Genitive). At each step, the results of a query can be examined to verify that it is returning the expected hits. When the process is complete, the corpus has effectively been annotated to identify (in this example) all genitive-case nouns. From this point on, the newly-defined `N-gen` object is now a first-class member of the corpus, and can be searched for and manipulated.

(6) Incrementally building up N-gen using a series of queries

- `//N[@lemma like "%ju" and @form like "%-ja"]`
- `//N[@lemma like "%l" and @suffix like "-la"]`
- `//N[@lemma like "%n" and @suffix like "-na"]`
- `//N[@lemma like "%r" and @suffix like "-ra"]`
- `//N[@suffix like "%a-ke4"]`
- etc.

This approach can be further extended to build more complex query objects out of simpler ones. In (BuildingComplexQueries) we see how a higher-level structure, an ergative-case noun phrase (`NP-erg`), can be built from the results of lower-level queries. In this example, the definition of `NP-erg` depends on having already defined queries to identify ergative-case nouns (`N-erg`) and genitive-case nouns (`N-gen`). An ergative-case noun (typically indicated in writing by a suffixed 𒂊 ‹e›) is inherently also an ergative-case noun phrase, so `N-erg` is the first part of defining `NP-erg`. However an ergative-case noun phrase could also consist of a pair of nouns with the second one bearing both a genitive-case *-ak* and the ergative-case *-e* (manifested orthographically as 𒂊 ‹ke$_4$›).

(7) Building NP-erg from lower-level queries

- `N-erg` defined as `//N[@suffix = "-e"]`[1]
- `N-gen` defined as in (6)
- `NP-erg` as `N-erg`
- `NP-erg` also as `//N <-- N-gen[@suffix like "\%-ke4"]`

Once the proper set of queries to define `NP-erg` has been determined, a new level of hierarchy has effectively been added to the corpus. As mentioned previously, one of the deficiencies of the ETCSL corpus was that it lacked any levels of structure between the word and the entire document. The query-based approach attempts to remedy that deficiency. Defining other phrase types, such as verb phrases and clauses, will be somewhat more complex, but the same principles can be used.

4. Practical Example

This section gives a practical example of how the approach of building up query objects can be used to locate data for an actual problem in Sumerian morphosyntax. The problem is the question of so-called “dimensional infixes” which were studied by Gragg 1973.

1 Like the `N-gen` object defined in (BuildingQueries), the actual queries to locate ergative-case nouns would have to be considerably more complex.

As shown in (8), Sumerian verbs have a large number of prefixes. The prefixes of interest are the ones in the middle of the chain, which agree with nouns elsewhere in the sentence: dative case, comitative case, ablative, terminative, locative, and ergative cases. The other prefixes, such as MODAL and CONJ, can safely be ignored. One of the advantages of this approach, as opposed to fully annotating the corpus, is that the queries need refer only to morphemes which are of immediate interest. This is particularly convenient for prefixes like CONJ, which are the subject of a great deal of heated debate (Thomsen 1984,Michalowski 2004).

(8) Prefix chain of the Sumerian verb

$$\text{MODAL - CONJ - DAT - COM - } \left\{ \begin{matrix} \text{ABL} \\ \text{TERM} \end{matrix} \right\} \text{ - LOC - ERG - verb – ABS}$$

One of the questions explored by Gragg 1973 was how the prefixes on the verb correspond to the associated nouns in the sentence. For instance, in (9) there is a nice example of a pronoun with the terminative-case suffix *-še* (written with the ⟨ še_3 ⟩ sign), and as expected, the verb appears with a terminative-case prefix, written with the ⟨ ši ⟩ sign.

(9) Example of terminative-case agreement

saĝ-ki zalag-ga-ni ĝa$_2$*-a-***še**$_3$ *hu-mu-***ši***-in-zig*$_3$
forehead shining-3SG.POSS 1SG-TERM hu-mu-TERM-ERG.3SG-lift
'she lifted her radiant forehead to me'

As Gragg discovered, this type of agreement is not present all the time. Sometimes the verb has the prefix and there's no terminative-case noun. Sometimes there is an appropriate noun, but the verb has a different prefix or no prefix at all. In order to explore these questions, it is necessary to define appropriate queries for the problem.
Query definitions to accomplish this are shown in (10). The first query finds all terminative-case nouns, and the second one identifies verbs which have the terminative-agreement prefix. Queries for the terminative and other cases are summarised in the Appendix.

(10) Queries for locating terminative-case agreement

- `N-term` defined as `//N[@suffix like "%-ce3"]`
- `V-term` defined as `//V[@suffix like "%-ci"] or @prefix like "%-ce3"]`
- Ideally: `//S{N-term <- V-term}`
- Realistically: `//PARA{N-term <- V-term}`

Ideally the scope of matching should be restricted to within a sentence (or better yet, a clause), but, as mentioned above, the corpus does not contain sentence boundaries. The best approximation is to scope matches within a paragraph. This is probably adequate for the task, since we can look at all the results and throw away the ones which are not relevant (e.g. spurious instances of agreement between a noun in one sentence and a verb in another sentence).

This raises an important point, which is that for a task like this study of agreement morphology, recall (i.e., finding every single example of a phenomenon) is more important than precision. As long as the query's result set is manageable, it can be pruned down manually to the examples which are actually of interest.

5. Discussion

The query-based approach presented here provides an alternative when annotation is unavailable or impractical. Although it has been presented here in reference to Sumerian and Elamite, it should be equally applicable to other low-resource languages where annotated corpora are similarly unavailable.

The approach is problem-specific and theory-neutral. The queries only create annotations which are actually needed, which avoids getting drawn into philological arguments about other morphemes. This is particularly important in languages such as Elamite and Sumerian, where the morphology is often poorly understood and subject to debate.

The approach works particularly well for problems like the agreement-morphology research which prompted this effort. In this sort of problem, recall is much important than precision. Thus the approach can aim for 100\% recall and sacrifice a certain amount of precision.

Importantly for languages whose orthography poorly reflects their morphology, this approach tries to insulate the linguist from the peculiarities of orthography. In particular, the goal is to allow the linguist to search for morphemes rather than graphemes.

Future work with the ETCSL corpus involves trying to define query objects for higher-level structures such as verb phrases and clauses. These are expected to be more complex than the queries described so far, but the general approach should still apply.

The question of Sumerian "dimensional infixes" described in §4 is only one of four which will be explored with this approach. For Sumerian, the approach will also be used to investigate conjugation prefixes. For Elamite, it will be used to research possessive constructions as well as object agreement morphology.

6. Summary

By defining a library of reusable query objects, it is possible to get many of the advantages of annotation without actually having to annotate. This approach is not specific to the languages or corpora described here, but is equally applicable to any corpus which lacks the resources for manual annotation.

Appendix: Query definitions for Sumerian dimensional infixes

The terminology used here follows Thomsen 1984. Under a newer classification of the Sumerian case system given by Michalowski 2004, the "terminative" is referred to as the "allative" and the "locative-terminative" is referred to as "locative2".

`N-dative` defined as:
- `//N[@suffix like "%-ra"]`
- `//N[@lemma like "%-a" suffix like "%-ar"]`
- `//N[@lemma like "%-i" suffix like "%-ir"]`
- `//N[@lemma like "%-u" suffix like "%-ur"]`

`V-dative-1SG` defined as:
- `//V[@prefix like "%-a-"]`

`V-dative-2SG` defined as:
- `//V[@prefix like "%-ra-"]`

`V-dative-3SG` defined as:
- `//V[@prefix like "%-na-"]`

`V-dative-1PL` defined as:
- `//V[@prefix like "%-me-"]`

`V-dative-3PL` defined as:
- `//V[@prefix like "%-ne-"]`

`N-comitative` defined as:
- `//N[@suffix like "%-da"]`

`V-comitative` defined as:
- `//V[@prefix like "%-da-"]`
- `//V[@prefix like "%-di-"]`
- `//V[@prefix like "%-de3-"]`
- `//V[@prefix like "%-de4"]`

N-locative defined as:[2]

- //N[@suffix like "%-a"]
- //N[@lemma like "%b_" suffix like "%-ba"]
- //N[@lemma like "%c_" suffix like "%-ca"]
- //N[@lemma like "%d_" suffix like "%-da"]
- //N[@lemma like "%g_" suffix like "%-ga"]
- //N[@lemma like "%h_" suffix like "%-ha"]
- //N[@lemma like "%j_" suffix like "%-ja"]
- //N[@lemma like "%k_" suffix like "%-ka"]
- //N[@lemma like "%m_" suffix like "%-ma"]
- //N[@lemma like "%n_" suffix like "%-na"]
- //N[@lemma like "%p_" suffix like "%-pa"]
- //N[@lemma like "%r_" suffix like "%-ra"]
- //N[@lemma like "%s_" suffix like "%-sa"]
- //N[@lemma like "%t_" suffix like "%-ta"]
- //N[@lemma like "%z_" suffix like "%-za"]

V-locative defined as:

- //V[@prefix like "%-ni-"]

N-terminative defined as:

- //N[@suffix like "%-ce3"]

V-terminative defined as:

- //V[@prefix like "%-ci-"]

N-ablative defined as:

- //N[@suffix like "%-ta"]
- //N[@suffix like "%-da"]

V-ablative defined as:

- //V[@prefix like "%-ta-"]

N-locative-terminative defined as:

- //N[@suffix like "%-e"]

V-locative-terminative defined as:[3]

- //V[@prefix like "%bi2-"]

2 Like the genitive case, the locative /-a/ suffix often assimilates with a stem-final vowel. This necessitates a rather cumbersome set of queries, since there is no easy way to express this with the current state of LPath.

3 The canonical form of the locative-terminative is /e/, but it typically assimilates with a preceding prefix, making its orthographic manifestation rather erratic. The tentative queries here are based on Michalowski (2004).

- `//V[@prefix like "%im-ma-"]`
- `//V[@prefix like "%mu-ni-"]`
- `//V[@prefix like "%-ri-"]`
- `//V[@prefix like "%-ni-"]`
- `//V[@prefix like "%-di-"]`
- `//V[@prefix like "%-de3-"]`
- `//V[@prefix like "%-de4-"]`

References

Bird et al. (2000): Steven Bird, Peter Buneman, and Wang-Chiew Tan. 2000. Towards a query language for annotation graphs. In: *Second International Conference on Language Resources and Evaluation*, pp. 807-814.

Bird et al. (2001-2007): Steven Bird, Ewan Klein, and Edward Loper. 2001-2007. *Natural Language Processing in Python*. Philadelphia: University of Pennsylvania.

Bird et-al. (2005): Steven Bird, Yi-Chen, Susan Davidson, Haejoong Lee, and Yifeng Zheng. 2005. Extending XPath to support linguistic queries. In: *Proceedings of Programming Language Technologies for XML (PLANX)*, Long Beach, January. ACM, pp. 35-46.

Bird et-al. (2006): Steven Bird, Yi-Chen, Susan Davidson, Haejoong Lee, and Yifeng Zheng. 2006. Designing and evaluating an XPath dialect for linguistic queries. In: *22nd International Conference on Data Engineering (ICDE)*, Atlanta, April, pp. 52-61.

Black et-al. (1998-2006): J.A. Black, G.-Cunningham, J.-Ebeling, E.-Flückiger-Hawker, E.-Robson, J.-Taylor, and G.-Zólyomi. 1998-2006. The Electronic Text Corpus of Sumerian Literature. `http://www-etcsl.orient.ox.ac.uk/` (18.1.2008).

Clark and DeRose (1999): James Clark and Steve DeRose. 1999. XML Path language (XPath). `http://www.w3.org/TR/xpath` (18.1.2008).

Englund and Damerow (2000-2005): R.-K. Englund and Peter Damerow. 2000-2005. Cuneiform Digital Library Initiative. `http://cdli.ucla.edu/` (8.1.2008).

Gragg, Gene-B.. 1973. *Sumerian dimensional infixes*. Kevelaer: Butzon und Bercker.

Grillot-Susini et-al. (1993): F.-Grillot-Susini, C.-Herrenschmidt, and F.-Malbran-Labat. 1993. La version élamite de la trilingue de Behistun: une nouvelle lecture. *Journal Asiatique* 281: 19-59.

Hinz, Walther and Heidemarie Koch. 1987. *Elamisches Wörterbuch*. Berlin: D. Reimer.

König, Friedrich-Wilhelm. 1965. *Die elamischen Königsinschriften*. Beiheft (Archiv für Orientforschung); 16. Graz: Im Selbstverlage des Herausgebers [E. Weidner].

Lai and Bird (2006): Catherine Lai and Steven Bird. 2006. LPath+: A first-order complete language for linguistic tree query. Unpublished manuscript.

Lambert, Maurice. 1974. Deux textes élamites du IIIe millénaire. *Revue Assyriologique* 68: pp. 3-14.

Michalowski, Piotr. 2004. Sumerian. In: Roger-D. Woodard (ed.). *The Cambridge Encyclopedia of the World's Ancient Languages*. Cambridge: Cambridge University Press.

Molina, Manuel. 2002-. Base de Datos de Textos Neosumerios. `http://bdts.filol.csic.es` (8.1.2008).

Paper, Herbert-H.. 1954. Note préliminaire sur la date des trois tablettes élamites de Suse. In: *Mémoires de la Délégation en Perse,* volume 36, Paris, pp. 79-82.

Scheil, V. 1907. *Textes élamite-anzanites, troisième série,* volume 9 of *Mémoires de la Délégation en Perse*.

Scheil, V. 1911. *Textes élamite-anzanites, quatrième série,* volume 11 of *Mémoires de la Délégation en Perse*.

Scheil, V. 1917. Déchiffrement d'un document anzanite relatif aux présages. *Revue d'Assyriologie* 14: 29-59.

Smith, Eric J.-M. 2004. *Optimality Theory and Orthography: Using OT to Reconstruct Elamite Phonology*. M.A. forum paper, University of Toronto.

Smith, Eric J.-M. 2006. *A Unified Account of Elamite Class-markers*. Generals paper, University of Toronto.

Smith, Eric J.-M. 2007. Phonological reconstruction of a dead language using the Gradual Learning Algorithm. In: *Proceedings of the Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology, 28 June 2007,* pp. 57-64.

Thomsen, Marie-Louise. 1984. *The Sumerian language: an introduction to its history and grammatical structure*. Copenhagen: Akademisk Forlag (Mesopotamia: Copenhagen studies in Assyriology; v. 10).

Tinney and Karahashi (2003-2004): Steve Tinney and Fumie Karahashi. 2003-2004. Pennsylvania Parsed Corpus of Sumerian. `http://psd.museum.upenn.edu/ppcs/` (8.1.2008).

Vallat, François. 1996. La lettre élamite de l'Armenie. *Zeitschrift für Assyriologie* 87: 258-269.

Veldhuis, Niek. 2003. Digital Corpus of Cuneiform Lexical Texts. `http://cuneiform.ucla.edu/dcclt` (8.1.2008).