

LCTL Formats and Conventions Plan

Version 2.5
(September 11, 2006)

Linguistic Data Consortium
<http://projects ldc.upenn.edu/LCTL/>

1	Introduction.....	3
2	Processing Pipeline	3
3	Data.....	6
3.1	Raw Text.....	7
3.2	LTF	7
3.3	LAF.....	7
3.4	LLF	7
4	Tools	8
5	Other Deliverables	8
5.1	Grammatical Sketch.....	8
5.1.1	Contents	8
5.1.2	Format	9
5.2	Name Transliterator	9
6	Appendix I: DTDs.....	10
6.1	ltf.v1.2.dtd.....	10
6.2	laf.v1.1.dtd	10
6.3	llf.v1.1.dtd.....	11
7	Appendix II: Data Samples.....	11
7.1	LTF	11
7.2	LAF.....	12
7.3	LLF	12
8	Appendix III: Filename Conventions (all applicable languages)	13
8.1	SRC.....	13
8.2	LNG	13
8.3	ID_UNIQUE.....	14
8.4	YYYYMMDD	14
8.5	Extension.....	14
8.6	Thread/Subject	14
8.7	hh.....	14
8.8	partID	14
9	Appendix IV: LDC Resources	14
9.1	SimpleNET	15
9.1.1	SimpleNET Tokenized Text	15
9.1.2	SimpleNET Annotations.....	16

1 Introduction

This document describes the LCTL deliverables provided by the LDC. For each language, we will deliver:

- Encoding Converter
- Tokenizer
- Sentence Segmenter
- Raw Monolingual Text (in [Raw Text](#) format)
- Sentence Segmented and Tokenized Monolingual Text (in [LTF](#) format)
- Raw Parallel Text (in [Raw Text](#) format)
- Sentence Segmented, Tokenized and Aligned Parallel Text (in [LTF](#) format)
- Lexicon (in [LLF](#) format)
- POS Tagger (and Tagset)
- POS-Tagged Text¹ (in [LTF](#) format)
- Morphological Analyzer (and Tagset)
- Morphologically-Analyzed Text² (in [LTF](#) format)
- Named Entity Tagger (and Tagset)
- Named Entity-Annotated Text (in [LAF](#) format)
- Name Lists
- Name Transliterator
- Grammatical Sketch

2 Processing Pipeline

The pipeline represents the stages of transformation that the data will undergo in the process of generating the multiple formats and annotations required in the 2005-06 LCTL resources packages.

The pipeline is designed to enforce a number of LCTL assumptions³:

1. There will be a consistent definition of “word” across all resources for each language:
 - a. Monolingual Text
 - b. Parallel Text
 - c. POS-Tagged Text
 - d. Named Entity-Annotated Text
 - e. Morphologically-Analyzed Text
 - f. Lexicon

¹ Selected languages only.

² Selected languages only.

³ In fact, these assumptions rarely hold in the found resources upon which LCTL relies. Early LoDL/LCTL releases, in order to exploit found resources, made deliveries in which some assumptions were not satisfied (with LCTL community agreement, of course). With time and increased capacity, many found resources are being processed to satisfy these assumptions and to inter-operate.

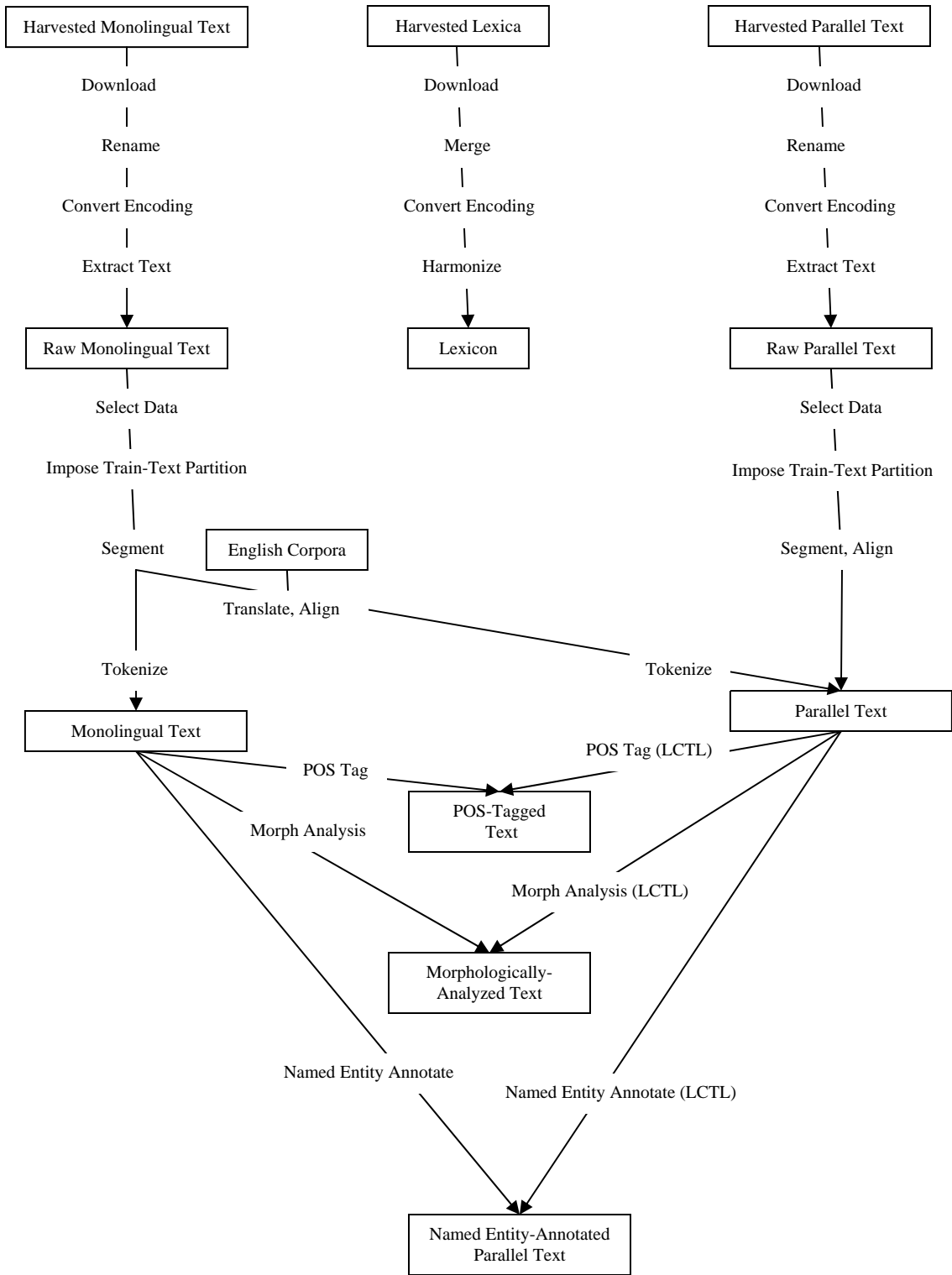
2. The source data and human-readable Raw Text will be provided.⁴
3. All major deliverables will be mutually-compatible.
4. There will be a consistent definition of training and evaluation data for at least⁵:
 - a. Monolingual Text
 - b. Parallel Text
 - c. Lexicon
 - d. Named Entity-Annotated Text

The processing pipeline represents the workflow under ideal circumstances. Exigencies of the language and corpus may require variations from this ideal. For example, the actual scripts and transformations will vary from language to language. Refer to the package documentation for specifics. Especially toward the bottom, the deliverables are highly inter-dependent. In many cases, multiple iterative loops through pieces of this pipeline will be required to obtain the required coverage.

Also note that many pieces of this pipeline will be jump-started with the help of found-resources. This practice will undoubtedly motivate small changes in the workflow described here.

⁴ The UTF8-encoded text extracted from original sources may include minimal XML tagging as needed to support processing and annotation, but will generally be kept in human-readable form, such that XML tags are easy to ignore or remove from the data.

⁵ Wherever possible, the Train/Eval partition will be established on the basis of source epochs.



The pipeline contains the following steps:

1. Harvest Monolingual Text and Parallel Text from the Internet:
 - a. Download the files.
 - b. Rename the files.
 - c. Convert the files to UTF-8.
 - d. Extract the usable text (LCTL and English) from the files.
2. Select Data for downstream processes.
3. Impose a monotonic Train/Eval partition.
4. (Sentence-)Segment the Monolingual and Parallel Text.
 - a. Train statistical models where simple rules do not work.
 - b. Verify the accuracy.
5. Segment-Align the found Parallel Text.
6. Translate selected LCTL Monolingual text (to become Parallel Text) into English. Also translate the English Corpora into the LCTL.
 - a. Text should be segmented, but not yet tokenized.
 - b. Translations will be segmented and segment-aligned by the translator.
 - c. Translations will be tokenized when they are returned to the LDC.
7. Tokenize the Monolingual and Parallel Text.
 - a. Train statistical models where simple rules do not work.
 - b. Verify accuracy.
8. Build Lexica.
 - a. Harvest, download, merge, and harmonize found dictionaries.
 - b. Measure lexical coverage.
 - c. Extend lexical coverage in frequency order of entries.
9. Train Morphological Analyzer.
10. Perform Morphological Analysis on select Monolingual and Parallel Text.
11. Train POS Tagger.
12. Perform POS Tagging on select Monolingual and Parallel Text.
13. Train Named Entity Tagger.
14. Perform Named Entity Annotation on select Monolingual and Parallel Text.

3 Data

With the exception of Raw Text, which will be delivered as it is found, all text data will be delivered in the Unicode UTF-8 encoding.⁶

LDC will harvest an enormous variety of data for the purposes of creating LCTL language packs. There will be nearly as many data formats as found resources.

Most data will contain both usable LCTL text and extraneous markup (e.g., HTML, PDF, SGML, MS Word, etc).

⁶ Text may instead be delivered in a reasonable substitute encoding, where UTF-8 is not possible or not desirable.

Each format is described below. We will extend the set of attributes as needed for particular languages; see the package documentation. For a complete DTD, see [Appendix I: DTDs](#).

3.1 Raw Text

These documents will be human-readable, containing minimal or no XML markup (e.g., <DOC> and <P> tags).

This Raw Text is the result of manually extracting contentful text from Harvested Text. As such, it will not be explicitly relatable to the original Harvested Text. The potential for PDF, GIF, JPG and other impenetrable formats to serve as a major source of text — especially for “non-digital” or “newly-digital” languages, such as Yoruba — is a major motivating factor in this decision.

3.2 LTF

Data in the LCTL Text Format will be sentence-segmented and tokenized. Additionally, a subset will have tokens bearing attributes indicating their POS or Morphological Analysis.

This format will serve as the format for most of the major LCTL deliverables:

1. (Segmented and Tokenized) Monolingual Text
2. (Segmented, Tokenized and Segment-Aligned) Parallel Text
3. POS-Tagged Text
4. Morphologically-Analyzed Text

Please see [ltf.v1.2.dtd](#) and an [LTF data sample](#).

3.3 LAF

The LCTL Annotations Format is used for annotations that span multiple tokens, which are less felicitously represented as attributes of individual tokens. LAF will be used only for Named Entity-Annotated Text for first-year deliveries, but it could also be used for NP-Chunking or similar annotations in future deliveries.

Please see [laf.v1.2.dtd](#) and an [LAF data sample](#).

3.4 LLF

The LCTL Lexicon Format is used for our Lexicons. The elements available to each entry in the Lexicon will be WORD, STEM, MORPH, MORPH_CLASS, ANIMACY,

POS, and GLOSS. Other languages may require additional attributes. See the package documentation for the actual attributes used.

Please see [llf.v1.2.dtd](#) and an [LLF data sample](#).

4 Tools

Each of the LCTL's deliverable tools:

1. Works with the LCTL's deliverable data⁷;
2. Includes all non-standard Libraries and Modules; and
3. Respects the Train-Test Partition when statistical modeling is used.

5 Other Deliverables

5.1 *Grammatical Sketch*

5.1.1 Contents

1. Introduction to the Language
 - a. Dialectology
 - b. Description of the Issues of Data in the Wild
2. Overview of Phonology and Orthography for:
 - a. Basics of Pronunciation
 - b. Encoding Conversion
 - c. Name Transliteration
 - d. Tokenization
 - e. Sentence Segmentation
 - f. Tagging
3. Inflectional Morphology (for parsing)
 - a. Form
 - b. Distribution
 - c. Semantics (not required)
4. Productive Derivational Morphology, e.g.,
 - a. Nominalized Relative Clauses
 - b. -ize → +VERB
5. Common Cliticizations
6. Commonly Misspelled Words / Spelling Irregularities
7. Nativization of Foreign Words and Names
8. Lists of Irregular Forms
9. Syntax:
 - a. Specification of Core NP
 - b. Basic Clause Structure

⁷ In early deliveries, minimal conversion will sometimes be necessary. Where applicable, conversion scripts will be provided.

- c. Major Syntactic Constructions
- d. Omissible Arguments
- 10. Other (optional)
 - a. Abbreviation Conventions
 - b. Special Sub-grammars as Appropriate (e.g., Name Grammar)

5.1.2 Format

- 1. Human Readable
- 2. UTF-8
- 3. Tabular Form
 - a. Controlled Vocabulary/ Consistent Representation⁸
 - b. Tagsets should be compatible with POS Annotations
 - c. Tagsets should be compatible with Morph Analysis
- 4. Links to:
 - a. Lexicon
 - b. Name Grammar

5.2 Name Transliterator

The Name Transliterator will take an input form in the LCTL's native script and output an official (or another likely) transliteration.

Optimally, the Name Transliterator will also output alternate forms and the frequency, weight, or ordering of those multiple forms.

⁸ Specific choices will vary from language to language. Details will be provided in the package documentation.


```

<!ELEMENT DOC (ANNOTATION)+ >
<!ATTLIST DOC
    id ID #REQUIRED
    lang CDATA #REQUIRED >

<!ELEMENT ANNOTATION (EXTENT) >
<!ATTLIST ANNOTATION id ID #REQUIRED
    task CDATA #REQUIRED
    type CDATA #REQUIRED
    start_token CDATA #REQUIRED
    end_token CDATA #REQUIRED >

<!ELEMENT EXTENT (#PCDATA) >
<!ATTLIST EXTENT
    start_char CDATA #IMPLIED
    end_char CDATA #IMPLIED >

```

6.3 *lft.v1.1.dtd*

```

<!ELEMENT LCTL_LEXICON (ENTRY*) >
<!ATTLIST LCTL_LEXICON lang CDATA #REQUIRED
    version CDATA #IMPLIED
    author CDATA #IMPLIED
    encoding CDATA #IMPLIED >

<!ELEMENT ENTRY
(WORD+, STEM*, MORPH*, MORPH_CLASS*, ANIMACY*, (POS, GLOSS)+) >
<!ATTLIST ENTRY id ID #REQUIRED >

<!ELEMENT WORD (#PCDATA) >
<!ELEMENT MORPH (#PCDATA) >
<!ELEMENT MORPH_CLASS (#PCDATA) >
<!ELEMENT ANIMACY (#PCDATA) >
<!ELEMENT STEM (#PCDATA) >
<!ELEMENT POS (#PCDATA) >
<!ELEMENT GLOSS (#PCDATA) >

```

7 Appendix II: Data Samples

7.1 LTF

```

<?xml version="1.0"?>
<!DOCTYPE LCTL_TEXT SYSTEM "lft.v1.2.dtd">
<LCTL_TEXT lang="THA" source_file="KRU_THA_20051003.0053.txt"
source_type="web_news" author="LDC" encoding="UTF-8">
<DOC id="KRU_THA_20051003.0053" lang="THA">
<TEXT>
<SEG id="KRU_THA_20051003.0053-1" start_char="11" end_char="50">
<ORIGINAL_TEXT>ชาติมุสลิมหนน ส่งเสริมลงทุน ชุมชนอิสลาม</ORIGINAL_TEXT>
<TOKEN id="KRU_THA_20051003.0053-1-1" start_char="11"
end_char="14">ชาติ</TOKEN>
<TOKEN id="KRU_THA_20051003.0053-1-2" start_char="15"
end_char="20">มุสลิม</TOKEN>
<TOKEN id="KRU_THA_20051003.0053-1-3" start_char="21"
end_char="24">หนน</TOKEN>

```

```

    <TOKEN id="KRU_THA_20051003.0053-1-4" start_char="26"
end_char="33">ส่งเสริม</TOKEN>
    <TOKEN id="KRU_THA_20051003.0053-1-5" start_char="34"
end_char="38">ลงทุน</TOKEN>
    <TOKEN id="KRU_THA_20051003.0053-1-6" start_char="40"
end_char="44">ชุมชน</TOKEN>
    <TOKEN id="KRU_THA_20051003.0053-1-7" start_char="45"
end_char="50">อิสลาม</TOKEN>
</SEG>
...
</TEXT>
</DOC>
</LCTL_TEXT>

```

7.2 LAF

```

<?xml version="1.0"?>
<!DOCTYPE LCTL_ANNOTATIONS SYSTEM "laf.v1.1.dtd">
<LCTL_ANNOTATIONS lang="THA"
lctl_text_file="AFP_THA_20030320.0722.ltf.xml" encoding="UTF-8">
<DOC id="AFP_THA_20030320.0722" lang="THA">
  <ANNOTATION id="AFP_THA_20030320.0722-NE1" task="NE" type="LOC"
start_token="AFP_THA_20030320.0722-1-24"
end_token="AFP_THA_20030320.0722-1-28">
    <EXTENT>ออกกลาง</EXTENT>
  </ANNOTATION> ...
</DOC>
</LCTL_ANNOTATIONS>

```

7.3 LLF

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LCTL_LEXICON SYSTEM "llf.v1.1.dtd">
<LCTL_LEXICON lang="THA" version="1.0" author="LDC">
  <ENTRY id="LEX-THA-00000100">
    <WORD>'ะ' ประวิสรรชนีย์</WORD>
    <POS>V</POS>
    <GLOSS>write the vowel</GLOSS>
  </ENTRY>
  <ENTRY id="LEX-THA-00000200">
    <WORD>( หนึ่ง ) ทุ่ม </WORD>
    <POS>ADV</POS>
    <GLOSS>19:00 h; 7 p.m. </GLOSS>
  </ENTRY>
  <ENTRY id="LEX-THA-00000300">
    <WORD>( หนึ่ง ) หมื่น ห้า พัน ห้า ร้อย </WORD>
    <POS>NONE</POS>
    <GLOSS>fifteen thousand; five fundred </GLOSS>
  </ENTRY>
  <ENTRY id="LEX-THA-00000400">

```

```

    <WORD>( หนึ่ง ) โมงเช้า </WORD>
    <POS>ADV</POS>
    <GLOSS>07:00 h; 7 a.m. </GLOSS>
</ENTRY>
<ENTRY id="LEX-THA-00000500">
    <WORD>( ขาย ) คลอง</WORD>
    <POS>ADV</POS>
    <GLOSS>sold-out</GLOSS>
</ENTRY>
</LCTL_LEXICON>

```

8 Appendix III: Filename Conventions (all applicable languages)

Filenames used in LCTL project may be composed of the following elements:

- SRC,
- LNG,
- ID_UNIQUE,
- YYYYMMDD,
- Extension,
- Thread/Subject (optional),
- hh (optional) and
- partID (optional).

8.1 SRC

This is the source of the file. It could be the name of the news source (e.g., BBC), the domain name of a newsgroup (e.g., groups.google.com), or the name of a blog (e.g., JUANCOLE). The actual sources and their codes will be listed in the package documentation.

8.2 LNG

This is the language the file is in. Each language is indicated using Ethnologue's 3-letter code. Ethnologue's language index is here:

http://www.ethnologue.com/language_index.asp

Ethnologue's language code index is here:

http://www.ethnologue.com/language_code_index.asp

The codes for languages for which the LCTL is currently developing language packs are:

Language	Code
Bengali	<i>BEN</i>
English	<i>ENG</i>
Hindi	<i>HIN</i>
Hungarian	<i>HUN</i>
Panjabi	<i>PAN</i>
Tagalog	<i>TGL</i>

Tamil	<i>TAM</i>
Thai	<i>THA</i>
Tigrinya	<i>TIR</i>
Urdu	<i>URD</i>
Uzbek	<i>UZN</i>
Yoruba	<i>YOR</i>

8.3 ID_UNIQUE

This is number in combination with the date and source uniquely identifies the file.

8.4 YYYYMMDD

This is the date: the year, month, and date, e.g., 20060817. This date may be the date of publication or the date of harvest.

8.5 Extension

The extension indicates the file type. For example, an extension of .xml indicates that the file is XML.

Extensions used in the LCTL project include:

- .txt ([Raw Text](#))
- .ltf.xml ([LTF](#))
- .laf.xml ([LAF](#))
- .llf.xml ([LLF](#))

8.6 Thread/Subject

This can be any type of string, e.g., an ID number with a text description. This is often either provided in the source HTML file or assigned to the file during the harvesting process.

8.7 hh

This is the time, either of publication or harvest. It is in 24-hour format, e.g., 00-23 is 12:23 a.m.

8.8 partID

This is used only when a particular post or article is extracted. It is a four-digit number. It can also be used when a broadcast transcript is divided into sentences.

9 Appendix IV: LDC Resources

The LCTL project's website is at:

<http://projects ldc.upenn.edu/LCTL/index.html>.

The LCTL documents all resources found for each language on “harvest pages.” You can view a list of harvest pages at:

<http://lodl ldc.upenn.edu/>.

The harvest page for each language can be found at:

http://lodl ldc.upenn.edu/LCTL/Language_harvest.html.

Note that “Language” here is a variable for “Yoruba,” “Panjabi,” etc.

The LCTL’s DTDs will be published online at:

<http://projects ldc.upenn.edu/LCTL/DTD>

The LCTL’s specifications for translation and Named Entity Annotation are at:

<http://projects ldc.upenn.edu/LCTL/Specifications/>.

9.1 SimpleNET

We have distributed Windows versions of this simpleNET annotation tool as part of an early LoDL/LCTL tools package:

http://projects ldc.upenn.edu/LCTL/Tools/SimpleNET_20060315.zip

There is also a self-installing version that is a little older:

http://projects ldc.upenn.edu/LCTL/Tools/LDC_LoDLTools.exe

SimpleNET uses two special formats: SimpleNET Tokenized Text and SimpleNET Annotations. We describe these two formats below only for those using the SimpleNET tool; these are not delivery formats.

9.1.1 SimpleNET Tokenized Text

The text document will consist of an XML header followed by the token stream from the raw UTF-8 text document.

Each token is displayed on its own line. A blank line indicates a paragraph or segment boundary. A designated symbol (here ‘|’) is used to indicate the absence of white space between adjacent tokens.

```
<TXTSTREAM name="ABC20001013.1830.0675.txt" >
```

ஆனால்

இந்த
கூட்டத்தின்
அவசரம்
|,
நேற்றைய
கலவரத்திற்குப்பின்
இஸ்ரேலியர்கள்
மற்றும்
பாலஸ்தீனர்கள்
அடைந்த
சீற்றத்திற்கு
நிகரானதாகும்
|. .
...
</TXTSTREAM>

9.1.2 SimpleNET Annotations

This text document consists of standoff annotations, to be used with SimpleNET Tokenized Text data described above. Each file will consist of the path to the SimpleNET Tokenized Text file and a list of Named Entities with the numbers of their associated tokens.

```
path_to_tokenized_text_file  
...  
47:47 PER  
49:49 PER  
...
```