

# Assignment 3: Morphological Analysis and Generation

Course: Computational Linguistics - 1

Deadline: February 27th, 2025 — 23:59

## 1 General Instructions

1. The assignment must be implemented using Apertium's lttoolbox.
2. Submitted assignment must be your original work. Please do not copy from any source.
3. Before getting started, ensure that Apertium and lttoolbox are properly installed.
4. Points distribution is provided for each section beforehand to avoid any confusion.
5. A single .zip file needs to be uploaded to the course portal.
6. Your grade will depend on correctness of paradigms, dictionaries, proper documentation and clarity of implementation.

## 2 Introduction

Apertium is an open-source platform for building machine translation systems, and lttoolbox is its morphological analyzer/generator component. This assignment focuses on building a basic morphological analyzer and generator using Apertium's lttoolbox. You will work with paradigms, create a dictionary, and implement morphological analysis and generation.

### 2.1 Basic Concepts

Morphological analysis breaks words into their base form (lemma) and grammatical features.

**Example:**

"walking" → walk (lemma) + present participle (feature)

Input word "walking"

↓

Morphological Analysis

↓

^walking/walk<vblex><gerund>\$

The reverse process (generation) creates word forms from lemmas and features.

### 3 Task 1: Building Paradigm [40 points]

A paradigm is a pattern showing how words change form.

**Example:** For verb "walk"

```
<pardef n="walk__vblex">
<!-- Present tense forms -->
<e>
  <p>
    <l>walk</l>
    <r>walk<s n="vblex"/><s n="pres"/></r>
  </p>
</e>
<e>
  <p>
    <l>walks</l>
    <r>walk<s n="vblex"/><s n="pres"/><s n="p3"/><s n="sg"/></r>
  </p>
</e>
<!-- Past tense form -->
<e>
  <p>
    <l>walked</l>
    <r>walk<s n="vblex"/><s n="past"/></r>
  </p>
</e>
</pardef>
```

This shows how "walk" changes form in different contexts:

walk → present tense  
walks → present tense, 3rd person singular  
walked → past tense

Create comprehensive paradigms for different word classes that demonstrate various morphological features. Your paradigms should include:

- Verb paradigms showing [25 Points]:
  - Tense variations (present, past, future) [8 points]
  - Aspect variations (simple, progressive, perfect) [9 Points]
  - Person and number agreement (1st, 2nd, 3rd person; singular/plural) [8 Points]
- Noun paradigm showing [10 Points]:
  - Number variations (singular/plural)
  - Case markers (if applicable in your chosen language)
- Documentation and organisation [5 Points]

## 4 Task 2: Building a Dictionary [25 points]

Create a morphological dictionary that includes:

- Minimum 20 entries across different word classes [15 points]:
  - At least 8 verbs [7 points]
  - At least 8 nouns [5 points]
  - At least 4 adjectives [3 points]
- Each entry must include [7 points]:
  - Lemma (base form)
  - Part of speech
  - Reference to appropriate paradigm
- Documentation and organization [3 points]

**Example:**

```
<section id="main" type="standard">
  <e lm="walk"><i>walk</i><par n="walk__vblex"/></e>
  <e lm="book"><i>book</i><par n="book__n"/></e>
  <e lm="quick"><i>quick</i><par n="quick__adj"/></e>
</section>
```

## 5 Task 3: Implementation and Testing using lttoolbox [35 points]

Implement and demonstrate the following using your paradigms and dictionary:

- Morphological Analysis [18 Points]:
  - Create an analyzer that can break down words into their component morphemes [10 Points]
  - Test with at least 10 different word forms [4 Points]
  - Document the analysis process and results [4 Points]
- Morphological Generation [17 points]:
  - Create a generator that can produce correct word forms from lemmas and morphological features [9 Points]
  - Generate at least 10 different forms for various lemmas [4 Points]
  - Verify the correctness of generated forms and document them. [4 Points]

**Example Analysis:**

**Input:** "walking"

**Expected Output:**

```
^walking/walk<vblex><gerund>$
```

**Example Generation:**

**Input:**

"walk<vblex><past>"

**Expected Output:**

^walked\$

## 6 Submission Guidelines

Submit a zip file named <roll\_number>\_assignment3.zip containing:

- Source files (.dix)
- Compiled binary files (.bin)
- Test files with sample inputs and outputs
- README.md with:
  - Usage Instructions
  - Documentation containing Results and Analysis
  - Any assumptions or limitations

## 7 Resources

- LTToolbox Documentation:  
<https://wiki.apertium.org/wiki/Lttoolbox>
- Apertium Installation Guide:  
<https://wiki.apertium.org/wiki/Installation>
- Monodix basics - Apertium:  
[https://wiki.apertium.org/wiki/Monodix\\_basics](https://wiki.apertium.org/wiki/Monodix_basics)