BA2: Digital Korea

Week 3: Text Preprocessing Basics

Steven Denney

Korean Studies
Leiden University

February 16, 2026

Today's Agenda

1. Week 2 review & DataCamp check-in
2. Key concepts: Tokenization
3. Key concepts: Part-of-speech (POS) tagging
4. The preprocessing pipeline
5. Stopwords and filtering
6. Our custom preprocessing scripts
7. Demo: Preprocessing workflow in Orange Data Mining
8. Looking ahead

Review & Check-In

Week 2 Review & DataCamp Check-In

Last week we covered:

- The research process: deductive vs. iterative
- Three stages: discovery, measurement, inference
- Six principles of text analysis (GRS Ch. 2)
- Corpora and data sources – loaded our first corpus in Orange

DataCamp: Introduction to Text Analysis in R – Ch. 1

How did it go? Questions about wrangling text in R?

Preprocessing

What Is Preprocessing?

Preprocessing = transforming raw text into a format suitable for computational analysis.

Raw text is messy

- Punctuation, capitalization
- URLs, special characters
- Grammatical particles
- Different forms of the same word

Analysis needs structure

- Consistent units (tokens)
- Meaningful words only
- Reduced noise
- Comparable across documents

Key insight

Preprocessing decisions shape your results. There is no "neutral" pipeline.

Tokenization

Tokenization = breaking text into individual units (“tokens”).

A token is the basic unit of analysis. Usually a word, but could be:

- Words: `"The cat sat on the mat"` $\rightarrow$ `[The, cat, sat, on, the, mat]`
- Subwords / morphemes: common in Korean
- Characters
- Sentences
- N-grams (sequences of *n* tokens)

Why it matters

The way you tokenize determines what counts as a “word” in your analysis.

Levels of Tokenization in Korean

Korean can be tokenized at multiple levels:

Level	Example	Description
Sentence	한국어를 배웁니다	Full sentence
Eojeol	한국어를 / 배웁니다	Space-delimited units
Morpheme	한국어 + 를 / 배우 + ㅂ니다	Smallest meaningful units
Syllable	한 국 어 를 배 웁 니 다	Individual syllable blocks
Jamo	ㅎ ㅏ ㄴ ㄱㅜㄱ ㅇ ㅓ	Consonants & vowels

For text analysis

We typically tokenize at the **morpheme** level. This is what our preprocessing scripts do using the Kiwi library.

Tokenization in English vs. Korean

English: Relatively simple

- Split on spaces and punctuation
- "I ate lunch" $\rightarrow$ [I, ate, lunch]
- Some edge cases: "don't", "New York"

Korean: Much harder

- Agglutinative language
- Particles attach to words
- 먹었습니다 = eat + past + polite
- Space-splitting is not enough

Example

나는 점심을 먹었습니다

Space split: [나는, 점심을, 먹었습니다] – 3 tokens, still complex

Morpheme split: [나, 는, 점심, 을, 먹, 었, 습니다] – 7 tokens

Why Korean Needs Morphological Analysis

Korean is **agglutinative**: meaningful units (morphemes) stick together.

Without morphological analysis:

- 먹다, 먹었다, 먹습니다, 먹어요 = 4 different “words”
- But they all mean “eat”!
- Your word counts become fragmented and unreliable

With morphological analysis (using Kiwi):

- All reduce to the root morpheme 먹 (eat)
- Grammatical endings (다, 었다, 습니다) are separated out
- We can keep or discard specific parts as needed

This is why we use custom Python scripts

Orange’s built-in preprocessing doesn’t handle Korean morphology. Our scripts use **Kiwi** (`kiwipiepy`), a Korean morphological analyzer.

Part-of-Speech Tagging

Part-of-Speech (POS) Tagging

POS tagging = labeling each token with its grammatical role.

English example

"The **cat** **sat** on the **mat**"

The/DT cat/**NN** sat/**VBD** on/IN the/DT mat/**NN**

Korean example (Kiwi tags)

대통령은 경제 성장을 강조했다

대통령/NNG 은/JX 경제/NNG 성장/NNG 을/JKO 강조/NNG 하/VV 었/EP 다/EF

POS tags let us **selectively keep** only the word types that matter for our analysis.

Korean POS Tags We Use

Our scripts use Kiwi's tag set. The key tags:

Content words (usually keep)

Tag	Meaning
NNG	General noun (일반명사)
NNP	Proper noun (고유명사)
VV	Verb (동사)
VA	Adjective (형용사)
MAG	Adverb (일반부사)

Grammatical words (usually remove)

Tag	Meaning
JX	Particle (보조사)
JKO	Object marker (목적격)
EP	Verb ending (선어말)
EF	Final ending (종결)
NNB	Bound noun (의존명사)

Default in our scripts

We keep **NNG** and **NNP** (nouns only). You can add VV, VA, MAG depending on your analysis.

Choosing POS Tags for Your Analysis

Different research questions need different word types:

Goal	**POS Tags**	**Rationale**
Topic modeling	NNG, NNP	Focus on *what* is discussed
Sentiment analysis	NNG, NNP, VA	Add evaluative language
Action analysis	NNG, NNP, VV, VA	Add *what's happening*
Exploratory	NNG, NNP, VV, VA, MAG	Maximum information

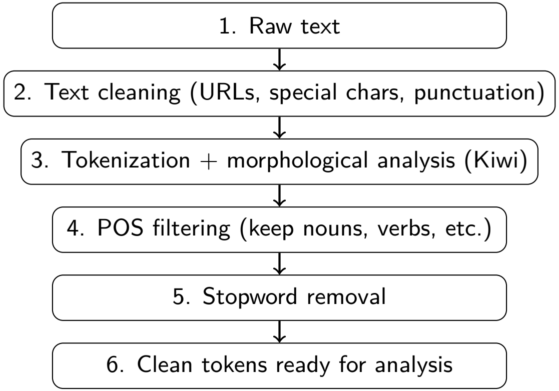
Tip

Start with nouns only. Add more POS categories if your results feel too sparse or if you need verbs/adjectives for your research question.

The Preprocessing Pipeline

The Preprocessing Pipeline

A typical preprocessing pipeline for Korean text:


1. Raw text
2. Text cleaning (URLs, special chars, punctuation)
3. Tokenization + morphological analysis (Kiwi)
4. POS filtering (keep nouns, verbs, etc.)
5. Stopword removal
6. Clean tokens ready for analysis

Step by Step: A Korean Example

Input: 이 제품은 정말 좋았습니다!

1. **Raw text:** 이 제품은 정말 좋았습니다!
2. **Cleaning:** 이 제품은 정말 좋았습니다 (punctuation removed)
3. **Tokenize + POS tag:**
   이/MM 제품/NNG 은/JX 정말/MAG 좋/VA 았/EP 습니다/EF
4. **POS filter** (nouns only): [제품]
   POS filter (nouns + adj): [제품, 좋]
   POS filter (nouns + adj + adv): [제품, 정말, 좋]
5. **Stopword removal:** (none of these are stopwords – keep all)
6. **Result:** 제품 좋 or 제품 정말 좋 (depending on POS config)

Stopwords & Filtering

What Are Stopwords?

Stopwords = words that appear frequently but carry little analytical meaning.

English stopwords

- the, a, an, is, was, are
- in, on, at, to, for
- and, but, or, not
- Very well-established lists

Korean stopwords

- 있다, 없다, 되다, 하다
- 것, 등, 및, 수
- 때, 더, 또, 즉
- Less standardized than English

Note

POS filtering already removes most grammatical noise (particles, endings). Stopword removal catches the remaining high-frequency, low-information content words.

Our Korean Stopwords List

We provide a curated stopwords file: `stopwords_ko.txt`

- 679 entries covering common particles, pronouns, adverbs, conjunctions, onomatopoeia, and special characters
- Available on the course website Data page and in the course repository
- Used via Orange's **Preprocess Text** widget (load as a custom stopword list)

Two layers of filtering

1. **POS filtering** (in our Python script): Keep only nouns, verbs, etc.
2. **Stopword removal** (in Orange or script): Remove remaining noise words

Together, these dramatically reduce noise and focus your analysis on meaningful content.

Our Custom Preprocessing Scripts

Why Custom Scripts?

Orange has built-in text preprocessing, but it doesn't handle Korean well:

- Orange's tokenizer splits on spaces – insufficient for Korean
- No morphological analysis for agglutinative languages
- 좋다, 좋았다, 좋습니다 treated as 3 unrelated words
- No Korean POS tagging built in

Our solution: Custom Python scripts that run *inside* Orange via the **Python Script** widget.

Available scripts (on Data page)

- `custom_preprocessing_annotations.py` – fully annotated version
- `custom_preprocessing_mac-users.py` – minimal, Mac
- `custom_preprocessing_windows-users.py` – minimal, Windows

The Annotated Script: What It Does

`custom_preprocessing_annotations.py` – the same script, with detailed comments explaining every step.

1. **Auto-installs** `kiwipiepy` (Korean NLP library) if needed
2. **Cleans text:** removes URLs, emails, @mentions, special characters
3. **Tokenizes** using Kiwi morphological analyzer
4. **POS filters:** keeps only specified word types (default: nouns)
5. **Removes stopwords** and very short tokens
6. **Outputs** a new column (`processed_text`) in your Orange table

Read the annotations!

The annotated script explains every line of code, every configuration option, and every design choice. Use it as a learning resource.

Key Configuration: POS Tags

In the script, you'll find:

```
POS_TAGS = [
    'NNG', # General nouns
    'NNP' # Proper nouns
    #'VV', # Verbs (uncomment to add)
    #'VA', # Adjectives (uncomment to add)
    #'MAG' # Adverbs (uncomment to add)
]
```

- Uncomment lines by removing the #
- Mind your commas!
- Start with nouns → add more if needed

Key Configuration: Text Column

Make sure the script knows which column contains your text:

```
TEXT_COLUMN = 'full_text' # Change to YOUR column name
```

- Check your CSV – what is the column header for the text?
- Common names: `full_text`, `text`, `content`, `body`
- Case-sensitive! `Full_Text` ≠ `full_text`

Mac vs. Windows: What's Different?

Mac version

- Auto-installs `kiwipiepy` with `--quiet` flag
- Generally "just works"
- Use: `custom_preprocessing_mac-users.py`

Windows version

- Assumes `kiwipiepy` is already installed
- May need manual install first via command line
- Use: `custom_preprocessing_windows-users.py`

Windows users: installing kiwipiepy

If the script errors on import, open a terminal and run:

```
pip install kiwipiepy
```

Then restart Orange and try again.

Orange Data Mining Demo

Demo: Preprocessing Workflow in Orange

Step 1: Load the corpus

1. Corpus widget → load presidential speeches CSV

Step 2: Run preprocessing script

2. Add Python Script widget → paste script
3. Set `TEXT_COLUMN` to your column name
4. Run – check the output log for "Processed *N* documents"

Step 3: Re-map and filter

5. Add a second Corpus widget → set `processed_text` as text
6. Add Preprocess Text → load `stopwords_ko.txt`

Step 4: Explore

7. Add Word Cloud to see the most frequent terms

Looking Ahead

For Next Week

Week 4: Text Preprocessing Practice

- Review and reinforcement of this week's concepts
- Morphological preparation and analysis
- Bag-of-words representation
- Term frequency measurements

Recommended reading:

- Grimmer, Roberts & Stewart – Chapter 5: Bag of Words
- Denny & Spirling (2018) – Text preprocessing for unsupervised learning

For Next Week

R Programming (required):

- DataCamp: Introduction to the Tidyverse – Chapter 1: Data Wrangling

Optional assignment (choose 1 or both):

1. Preprocess a text and create a word cloud **in Orange Data Mining**
2. Preprocess a text and create a word cloud **in RStudio** using a provided R script

Details will be posted on the Assignments page. **Orange tutorial:**

- Getting Started 16: Text Preprocessing