BA2: Digital Korea

Week 6 Assessment Study Guide — Covering Weeks 1–5

Dr. Steven Denney | Korean Studies, Leiden University | Spring 2026

Assessment format. The Week 6 assessment has two parts:

1. **Online quiz** (∼20 min) — multiple-choice questions covering concepts from Weeks 1–5.
2. **Hands-on task** (∼20 min) — download a small new corpus, load it into Orange, run it through the preprocessing stages, and produce a clean Word Cloud demonstrating success. Upload to your shared private GitHub repo: your `.ows` file, a saved image of the Word Cloud, and a short `.md` file explaining what research question you might ask with the data and what you would expect to find.

This guide covers everything you need to review.

Contents

1 Korean Language and Text Analysis

1.1 Why Korean Needs Special Tools

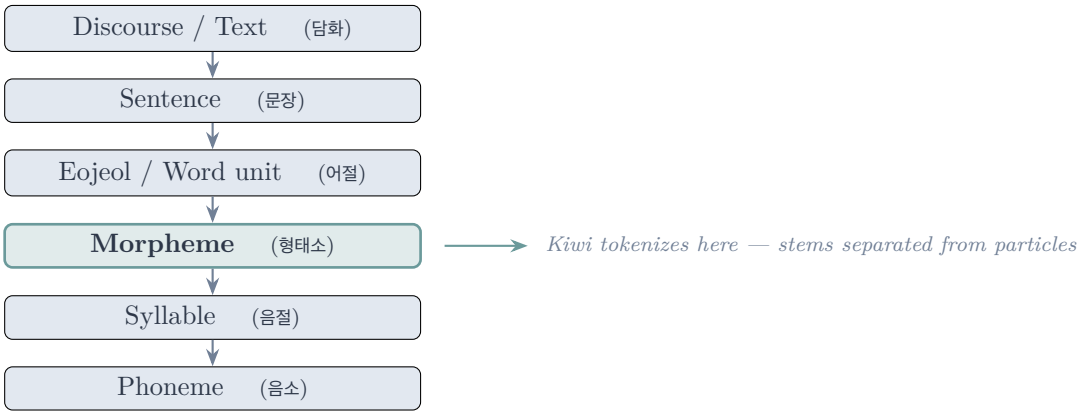
Korean is an **agglutinative** language: grammatical information is expressed by attaching particles, endings, and suffixes to word stems. Unlike English, where spaces reliably separate meaningful words, a single Korean "word" (orthographic unit) often contains multiple meaningful parts glued together.

Surface form	Decomposition	Meaning
경제를	경제 + 를	economy + object particle
경제의	경제 + 의	economy + possessive particle
경제가	경제 + 가	economy + subject particle
학생들이	학생 + 들 + 이	student + plural + subject

If we simply split Korean text on spaces (as we might for English), each of these forms would be counted as a *different word*, even though they all refer to the same concept. This is why we need **morphological analysis**.

1.2 Linguistic Levels

When we process text computationally, we choose a **unit of analysis**. The hierarchy of linguistic units below shows where that choice falls:



A **morpheme** (형태소) is the smallest meaningful unit of language. In Korean text analysis, we tokenize at the morpheme level because:

- It separates stems from particles and endings
- It reduces inflected forms to their base (dictionary) forms
- It produces cleaner, more meaningful tokens for counting and comparison

1.3 Morphological Analysis vs. Lemmatization

In English NLP, **lemmatization** reduces words to their dictionary form: *running*, *runs*, *ran* → **run**.

For Korean, the equivalent process is **morphological analysis** (형태소 분석). It does more than lemmatization — it decomposes each *eojeol* into its constituent morphemes and labels each one with a part-of-speech (POS) tag.

Surface form	Morphological analysis	What it means
배웁니다	배우/VV + ㅂ니다/EF	learn + formal ending
배워요	배우/VV + 어요/EF	learn + polite ending
배운다	배우/VV + ㄴ다/EF	learn + declarative ending

All three forms reduce to the same base form 배우다 ("to learn"). Without morphological analysis, they would be counted as three different words.

1.4 Kiwi and POS Tags

In this course, we use **Kiwi** (`kiwipiepy`), a Korean morphological analyzer. Kiwi breaks each word into morphemes and assigns POS tags. We then **filter by POS tag** to keep only the parts of speech we care about.

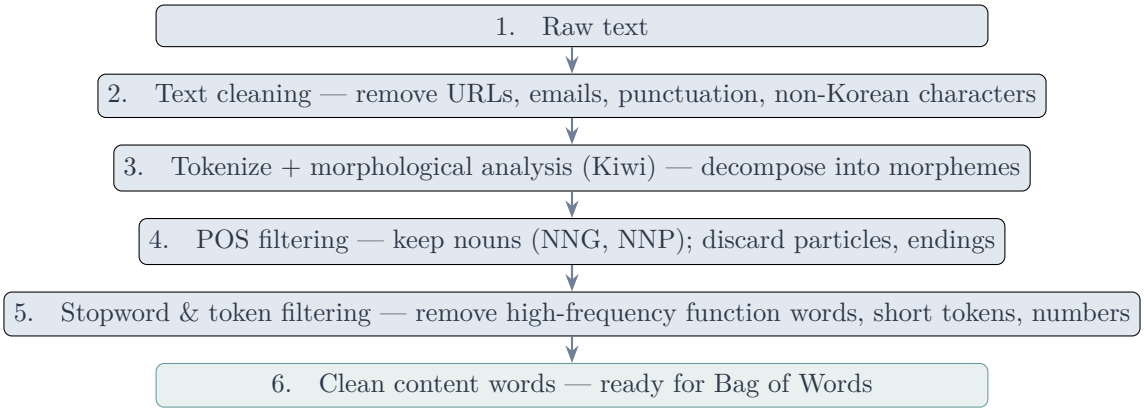
Tag	Category	Korean	Examples
NNG	Common noun	일반명사	사회, 경제, 학생
NNP	Proper noun	고유명사	서울, 한국
VV	Verb	동사	가다, 먹다
VA	Adjective	형용사	크다, 좋다
MAG	Adverb	부사	매우, 아주

> By default, our preprocessing script keeps only **NNG** and **NNP** (nouns). This is a common starting point for topic-oriented analysis. You can optionally include verbs and adjectives for richer analysis.

2 The Preprocessing Pipeline

Preprocessing transforms raw text into a clean, structured form that a computer can analyze. Every step has a purpose.

2.1 The Six Steps



2.2 What the Python Script Does

In Orange, our preprocessing happens in a **Python Script widget** that runs a custom script. You should understand what each part of that script does — not the Python syntax, but the *logic* of each step.

2.2.1 Step 1: Text Cleaning

Before morphological analysis, the script removes noise from the raw text:

- URLs (`http://...`), email addresses, @mentions
- Non-Korean, non-alphanumeric characters
- Extra whitespace

This gives Kiwi cleaner input to work with.

2.2.2 Step 2: Morphological Analysis (Kiwi)

Kiwi's `tokenize()` function takes each cleaned document and returns a list of morphemes, each with a base form and a POS tag. For example:

학생들이 경제를 배웁니다 ⟶ 학생/NNG, 들/XSN, 이/JKS, 경제/NNG, 를/JKO, 배우/VV, ㅂ니다/EF

2.2.3 Step 3: POS Filtering

The script keeps only morphemes whose POS tag matches our target list (by default: NNG, NNP). Everything else — particles, endings, suffixes — is discarded. From the example above, only 학생 and 경제 survive.

2.2.4 Step 4: Stopword and Token Filtering

Three additional filters:

- **Stopwords**: Common words that pass the POS filter but carry little meaning (있다, 없다, 하다, 것, etc.)
- **Minimum token length**: Single-character tokens are removed (often noise)
- **Number removal**: Purely numeric tokens are discarded

2.2.5 Step 5: Document Frequency Filtering

After tokenizing all documents, the script applies corpus-wide filters:

- **Minimum document frequency** (default: 10%) — words appearing in fewer than 10% of documents are dropped (too rare to reveal patterns)
- **Maximum document frequency** (default: 90%) — words appearing in more than 90% of documents are dropped (too common to be informative)

> **Don't confuse document frequency filtering with TF-IDF.** Document frequency filtering *removes* words entirely. TF-IDF *reweights* words — common words get lower scores but are still present.

2.2.6 Step 6: Output

The script adds a new column called `processed_text` to your data. Each cell contains a space-separated string of clean base-form tokens. This column is what you select in the second Corpus widget and connect to the **Bag of Words** widget.

3 From Words to Numbers

3.1 Key Terms

Term	Definition	Description	Think of it as...
BoW	Bag of Words	Represent text by word counts, ignoring word order and grammar	Cutting words out and dropping them in a bag
TF	Term Frequency	The count of a word in one document (can be normalized by document length)	One cell of the matrix
DF	Document Frequency	The number of documents in which a word appears	One column's spread across rows
DTM	Document-Term Matrix	The table (documents × words) that the BoW widget creates	The spreadsheet the computer "sees"
IDF	Inverse Doc. Freq.	$\log(N/\text{DF})$; measures how rare a word is across the corpus	High IDF = rare = informative
TF-IDF	TF × IDF	Weighting that highlights words frequent in a document but rare across the corpus overall	"Frequent here, rare elsewhere"

3.2 The Bag of Words Model

The Bag of Words (BoW) model represents each document as a vector of word counts. It deliberately **discards**:

- Word order ("the dog bit the man" = "the man bit the dog")
- Grammar and syntax
- Context and meaning

This is a drastic simplification — but it works surprisingly well for many tasks because *which* words a document uses is often enough to characterize it.

3.3 The Document-Term Matrix (DTM)

The DTM is the table that BoW produces. Each **row** is a document, each **column** is a word, and each **cell** is a count (or weight).

	경제	통일	평화	민주	...
Speech 1	12	3	5	0	...
Speech 2	0	8	7	1	...
Speech 3	5	0	0	9	...

> Most cells in a DTM are **zero** — this is called **sparsity**. Most words don't appear in most documents. This is normal and expected.

3.4 Matrix vs. DTM vs. Dataframe

Concept	What it is	Where you see it
Matrix	A grid of numbers. No labels, no column names — just rows and columns of values.	General math concept
DTM	A *specific* matrix: rows = documents, columns = terms, cells = counts or weights.	What the Bag of Words widget creates
Dataframe	A table with named columns of mixed types (text, numbers, dates).	Your CSV file; what the Corpus widget shows

Your CSV is a **dataframe**. After Bag of Words, it becomes a **DTM**. Both are tables, but the DTM is all numbers, built for computation.

3.5 TF-IDF Weighting

Raw word counts can be misleading. A word like 국민 ("the people") might appear frequently in *every* presidential speech, making it useless for distinguishing between presidents. TF-IDF addresses this.

- **TF** (Term Frequency): How often a word appears in *this* document
- **IDF** (Inverse Document Frequency): $\log(N/\text{DF})$ — how rare the word is across the corpus. Words that appear in many documents get a low IDF.
- **TF-IDF** = TF × IDF: Words that are frequent *here* but rare *elsewhere* get the highest scores.

> **Quick test.** If a word has *high TF* but *low TF-IDF*, what does that mean? It means the word is frequent in this document but also common across the corpus — it's not distinctive.

3.6 What the BoW Widget Does

The Bag of Words widget builds the DTM and gives you three controls that change the cell values:

1. **Count** — each cell is the raw integer count of a word in a document.
2. **IDF toggle** — multiplies each cell by the word's IDF weight. Words spread across many documents get down-weighted (e.g., 국민 appears in nearly every speech, so its IDF ≈ 0.19).
3. **Regularization** (labeled this way in Orange; this is **L2 normalization**) — scales each document row to unit length, removing the effect of document length differences. A 5,000-word speech and a 500-word speech become comparable.

> **How visualization widgets read the DTM.** Both the Word Cloud and Bar Plot display the **mean value across all documents** for each word. With Count, a word with 3,977 total occurrences across 749 speeches shows as ≈5.31 (not 3,977). Toggle IDF on and that value might drop to ≈1 because the word's high document frequency is down-weighted.

4 Descriptive Exploration in Orange

Before any deeper analysis, you should always **explore your data descriptively**. This validates your preprocessing, reveals corpus structure, and generates hypotheses.

4.1 The Toolkit

Widget	What it shows	When to use it
Word Cloud	Visual overview of frequent terms; size = relative frequency	First look at your data; preprocessing sanity check
Bar Plot	Exact term frequencies or TF-IDF scores with clear ranking	Comparing values precisely across terms
Concordance	Keywords in their original context (KWIC display)	Understanding how a word is actually used
Statistics	Per-document metrics (word count, unique words)	Spotting outliers and data quality issues
Select Rows	Filters corpus by metadata values	Creating subsets (by president, year, speech type)

4.2 Word Clouds vs. Bar Charts

Word clouds are useful for a **first look** — they show frequent terms at a glance and can catch leftover stopwords or preprocessing problems. But they are *imprecise*: size differences are hard to judge, the layout changes each time, and they can mislead when comparing groups.

Bar charts give you what word clouds cannot: exact values and clear ranking. Always follow up with bar charts for anything you want to claim or compare.

Both widgets display mean values from the DTM, so the numbers you see depend entirely on your BoW settings (Count, IDF, Regularization). Make sure you know which combination is active.

4.3 Concordance (KWIC)

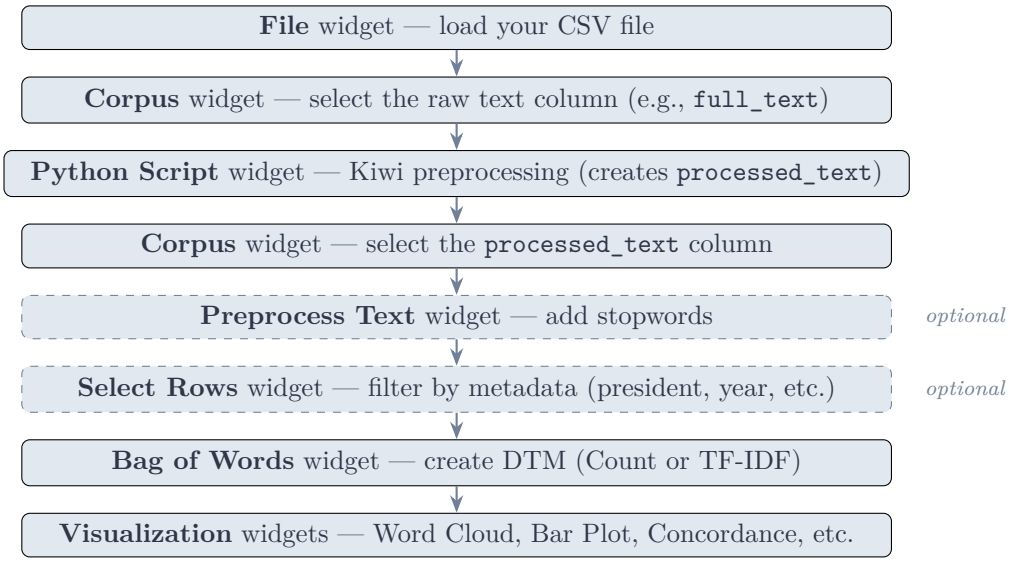
Concordance shows every occurrence of a keyword with its surrounding text — a bridge between quantitative and qualitative analysis. A word count tells you *how often*; concordance tells you *how* the word is used.

4.4 Select Rows for Subsetting

The **Select Rows** widget filters your corpus by metadata. Place it **before** Bag of Words to focus your analysis on a subset (e.g., one president's speeches). This means TF-IDF scores will be calculated relative to that subset only.

5 The Complete Orange Workflow

Putting it all together, a typical workflow in Orange follows this sequence:



Why two Corpus widgets? The first Corpus widget tells Orange which column contains your raw text. After the Python Script creates the `processed_text` column, the second Corpus widget tells Orange to use *that* column instead.

6 Self-Check Questions

Use these to test your understanding. If you can answer all of them confidently, you're well prepared.

6.1 Korean Language & Preprocessing

1. Why can't we simply split Korean text on spaces, the way we can (roughly) for English?
2. What is a **morpheme**? Where does it sit in the hierarchy of linguistic units?
3. What does Kiwi do, and why is it necessary for Korean text analysis?
4. What is the difference between **lemmatization** and **morphological analysis**?
5. Name two POS tags we keep by default and two types of morphemes we discard.
6. What do stopwords like 있다, 하다, and 것 have in common? Why remove them?
7. What is the purpose of **document frequency filtering**? How is it different from TF-IDF?

6.2 Bag of Words & TF-IDF

1. What information does the Bag of Words model **discard**?
2. What is the DTM? What do its rows, columns, and cells represent?
3. Why is the DTM mostly zeros?
4. If a word has high TF but low TF-IDF, what does that tell you?
5. A word appears in 748 out of 749 speeches. Is its IDF high or low? Why?
6. What is the difference between TF and DF?
7. When would you use L2 normalization, and why?

6.3 Descriptive Exploration

1. Why should you explore your data descriptively *before* deeper analysis?
2. What is one thing a word cloud shows well, and one thing it does poorly?

3. The Bar Plot widget shows different values depending on your BoW settings. Explain.
4. What does the Concordance widget show that word counts cannot?
5. If you place Select Rows *before* Bag of Words, how does that affect TF-IDF scores?

Focus on understanding the *why* behind each step, not just the *how*.

A Glossary of Key Terms

Agglutinative language A language (like Korean) where grammatical information is expressed by attaching particles, endings, and suffixes to word stems.

Bag of Words (BoW) A text representation model that counts word occurrences while discarding word order, grammar, and context. The foundational "text as data" representation used in this course.

Bar Plot An Orange widget that displays exact term frequencies or TF-IDF scores as a ranked bar chart.

Computational Text Analysis (CTA) The systematic use of computational methods to analyze text data. Sits at the intersection of computational social science and digital humanities.

Concordance (KWIC) A display of every occurrence of a keyword with its surrounding context (Keyword in Context). Bridges quantitative and qualitative analysis.

Corpus A structured collection of texts assembled for analysis. In Orange, also the name of the widget that designates which column holds the text data.

Dataframe A table with named columns of mixed types (text, numbers, dates). Your CSV file is a dataframe.

Document Frequency (DF) The number of documents in which a given word appears. Used to measure how widespread a word is across the corpus.

Document Frequency Filtering Removing words that appear in too few or too many documents. Unlike TF-IDF, this *removes* words entirely rather than reweighting them.

Document-Term Matrix (DTM) A table where rows are documents, columns are words, and cells are counts or weights. The output of the Bag of Words widget.

Eojeol (어절) A Korean orthographic word unit — the string of characters between spaces. May contain multiple morphemes.

Inverse Document Frequency (IDF) $\log(N/\text{DF})$; measures how rare a word is across the corpus. High IDF means rare and potentially informative.

Kiwi (`kiwipiepy`) A Korean morphological analyzer that decomposes words into morphemes and assigns POS tags.

L2 Normalization Scaling each document's row in the DTM to unit length, making documents of different lengths comparable. Labeled **Regularization** in Orange's Bag of Words widget.

Lemmatization Reducing words to their dictionary form (lemma). In English: *running* → *run*. For Korean, morphological analysis serves this role.

Matrix A grid of numbers with no labels or column names. A general mathematical concept; the DTM is a specific type of matrix.

Metadata Data about your data — variables like president name, date, speech type that describe each document but are not part of the text itself. Used for filtering and grouping.

Maximum Document Frequency A threshold (default: 90%) above which words are removed for being too common across the corpus to be informative.

Minimum Document Frequency A threshold (default: 10%) below which words are removed for being too rare to reveal patterns.

Morpheme (형태소) The smallest meaningful unit of language. Korean tokenization operates at this level, separating stems from particles and endings.

Morphological Analysis (형태소 분석) Decomposing words into their constituent morphemes and labeling each with a POS tag. The Korean equivalent of lemmatization, but more comprehensive.

NNG / NNP POS tags for common nouns

(일반명사) and proper nouns (고유명사), respectively. The default tags kept by our preprocessing script.

Normalization Adjusting values to a common scale. In this course: *TF normalization* adjusts for document length within term frequency; *L2 normalization* scales entire document vectors to unit length for fair comparison.

Part-of-Speech (POS) Tag A label assigned to each morpheme indicating its grammatical category (noun, verb, particle, etc.). Used to filter which morphemes to keep.

Preprocessing The process of transforming raw text into a clean, structured form suitable for computational analysis. Includes cleaning, tokenization, POS filtering, and stopword removal.

Select Rows An Orange widget that filters the corpus by metadata values (e.g., president, year). Used to create subsets for focused analysis.

Sparsity The property of a DTM where most cells are zero, because most words don't appear in most documents.

Statistics An Orange widget that reports per-document metrics such as word count and unique words. Useful for spotting outliers.

Stopwords High-frequency words that carry little analytical meaning and are removed during preprocessing (e.g., 있다, 하다, 것).

Term Frequency (TF) The count of a word in one document. Can be normalized by document length. Represents one cell of the DTM.

TF-IDF TF × IDF. A weighting scheme that highlights words that are frequent in a specific document but rare across the corpus overall.

Token An individual unit produced by tokenization. In Korean CTA, tokens are morphemes (base forms) extracted by Kiwi, not whole words.

Tokenization Breaking text into individual units (tokens). For Korean, this is done via morphological analysis rather than simple space-splitting.

Validation Checking that your computational results are meaningful and accurate. Includes verifying preprocessing output, inspecting results against domain knowledge, and not trusting numbers blindly.

Vector A one-dimensional array of numbers. In CTA, each document is represented as a vector of word counts or weights — one row of the DTM.

Weighting Transforming raw counts into scores that better reflect a word's importance. TF-IDF is the primary weighting scheme used in this course.

Word Cloud An Orange widget that shows a visual overview of term importance, where word size reflects the mean value across all documents. Values depend on BoW settings (Count, IDF, Regularization).