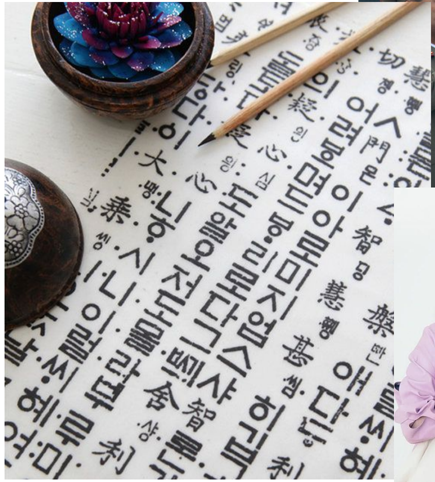
Ling 409
Final Project

By Samanvay Upadhyay, Jordan Day, and Milo Han

Introduction



- ~80 million native speakers primarily in North and South Korea
- Significant diaspora of speakers in USA, China, Japan, and Central Asia
- Part of the Koreanic language family; formerly thought to be related to Japonic, Mongolic, Tungusic, etc
- Head-final agglutinative morphology
- Most notable for being the subject of this project

Syntactic Generalizations (Task 1)

Syntactic Generalizations

We decided to test the rigidity of Korean SOV word order, particularly that the placement of the object is always before the verb.

Generalization: “If a verb has an object, the object will always be found before the verb.”

To test this generalization and our later verbal generalizations, we used the KAIST UD treebank (350k tokens) for Korean generated by Chun et al., 2018.

Code

For each verb, if it has an object, look at where the object is located relative to the verb

- If it is right before the verb, add to the *immediate* count
- If it is within 3 tokens before the verb, add to the *within 3* count
- If it is before the verb at all, add to the *before* count
- If it is after the verb, add to the *after* count
- These counts are not mutually exclusive.

```
for sent_idx, tokens in enumerate(sents):
    id_to_tok = {t['id']: t for t in tokens}
    dependents = defaultdict(list)
    for t in tokens:
        if t['head'] is not None and t['head'] in id_to_tok:
            dependents[t['head']].append(t)
    id_to_index = {t['id']: i for i, t in enumerate(tokens)}

    for i, tok in enumerate(tokens):
        if tok['upos'] == 'VERB':
            verb_instances += 1
            verb_id = tok['id']
            deps = dependents.get(verb_id, [])
            for dep in deps:
                if dep['deprel'] == 'obj':
                    obj_instances += 1
                    obj_idx = id_to_index.get(dep['id'])
                    verb_idx = i
                    if obj_idx is None:
                        continue
                    diff = verb_idx - obj_idx  # positive if object is to
verb's left
                    dist_counter[diff] += 1
                    if diff == 1:
                        obj_immediate += 1
                    if 1 <= diff <= 3:
                        obj_within3 += 1
                    if diff < 0:
                        obj_to_right += 1
                    if diff > 0:
                        obj_any_before += 1


                    lemma = tok['lemma'] if tok['lemma'] != '_' else
tok['form']
                    if len(examples[lemma]) < 5:
                        sent_form = ' '.join([t['form'] for t in tokens])
                        examples[lemma].append((dep['form'],
tok['form'], sent_form))

if obj_instances > 0:
    pct_immediate = obj_immediate / obj_instances * 100
    pct_within3 = obj_within3 / obj_instances * 100
    pct_after = obj_to_right / obj_instances * 100
else:
    pct_immediate = pct_within3 = 0.0
```

Analysis

From the corpus, we found that:

- 72.35% directly precede the verb
- 92.99% are within 3 tokens of the verb
- 100% are before the verb
- 0% come after the verb

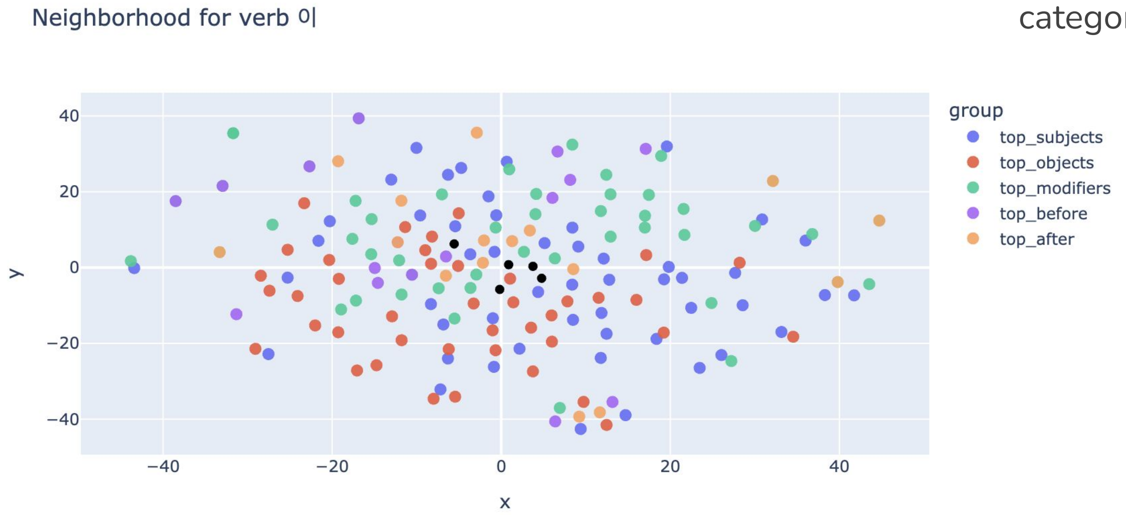
Our generalization that objects always come before their verb is thus justified by the data.

This holds true for every case in our data set, although the actual distance between the object and verb varies.

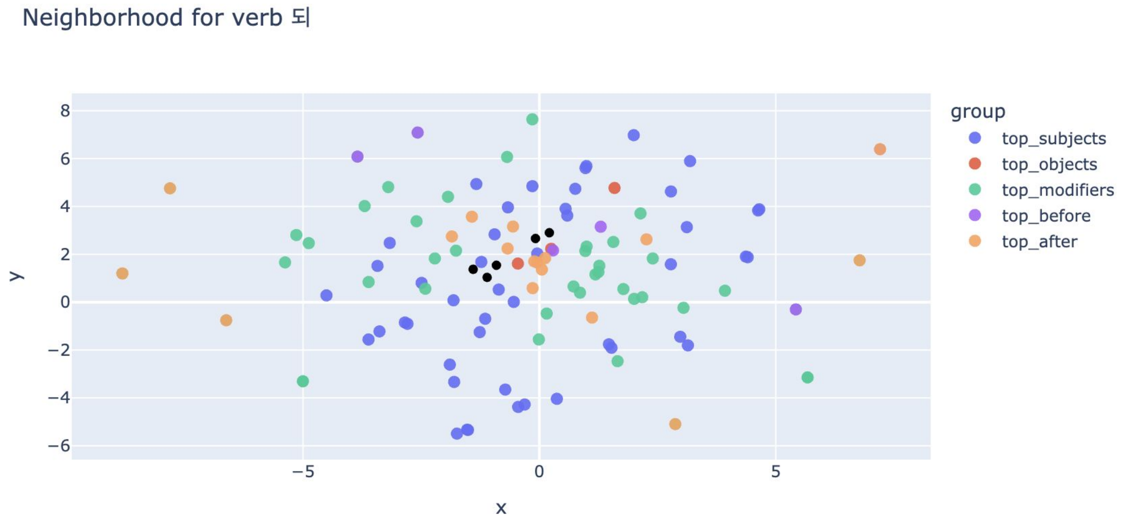
```
for sent_idx, tokens in enumerate(sents):
    id_to_tok = {t['id']: t for t in tokens}
    dependents = defaultdict(list)
    for t in tokens:
        if t['head'] is not None and t['head'] in id_to_tok:
            dependents[t['head']].append(t)
    id_to_index = {t['id']: i for i, t in enumerate(tokens)}

    for i, tok in enumerate(tokens):
        if tok['upos'] == 'VERB':
            verb_instances += 1
            verb_id = tok['id']
            deps = dependents.get(verb_id, [])
            for dep in deps:
                if dep['deprel'] == 'obj':
                    obj_instances += 1
                    obj_idx = id_to_index.get(dep['id'])
                    verb_idx = i
                    if obj_idx is None:
                        continue
                    diff = verb_idx - obj_idx  # positive if object is to
verb's left
                    dist_counter[diff] += 1
                    if diff == 1:
                        obj_immediate += 1
                    if 1 <= diff <= 3:
                        obj_within3 += 1
                    if diff < 0:
                        obj_to_right += 1
                    if diff > 0:
                        obj_any_before += 1


                    lemma = tok['lemma'] if tok['lemma'] != '_' else
tok['form']
                    if len(examples[lemma]) < 5:
                        sent_form = ' '.join([t['form'] for t in tokens])
                        examples[lemma].append((dep['form'],
tok['form'], sent_form))

if obj_instances > 0:
    pct_immediate = obj_immediate / obj_instances * 100
    pct_within3 = obj_within3 / obj_instances * 100
    pct_after = obj_to_right / obj_instances * 100
else:
    pct_immediate = pct_within3 = 0.0
```

Neighborhood Findings and Semantic Verb Generalizations (Task 2b)

* black dots are centroids of each category



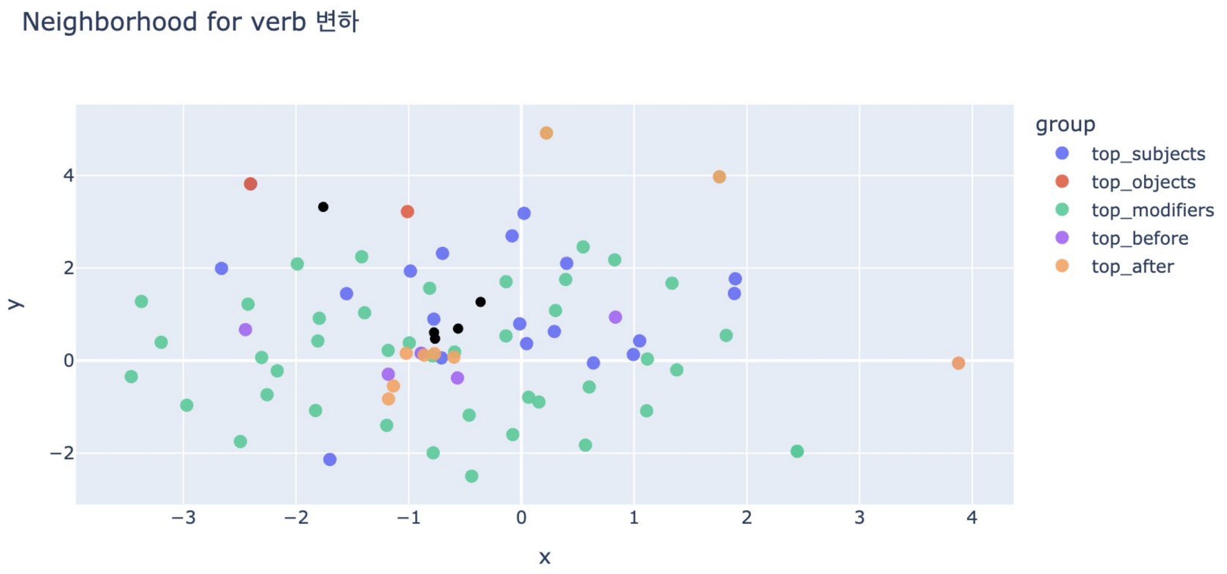
We tried mapping the neighborhood for one of our verbs to first verify that our results were as expected, but we didn't get much useful information from it. This is likely because the verb 이다 (*i.*) is the copula, so it has a very broad scope, appearing basically anywhere nouns are used.



Using a more specific verb 되다 (*twoi.*, "to become"), we got a slightly more clustered group, with lots of the meanings matching up to words that realistically would be used with our verb, so we were able to verify our code. We can also determine that since the range of the scales is smaller on this graph, there is a more specific use-case for this verb as opposed to the previous one, which has a broad usage.

Looking more at the verb 되다 (*twoi.*, “to become”), we see that the top subject is ‘것+이’ (*kes.i*, “thing”), the top object is ‘아이+들+을’ (*a.i.dul.ul*, “children”) (though it was not often used with objects), top modifier (and top before) is ‘안’ (*an.*, “not”), top after is ‘.’ (“.”). We can see that while all the top sets are pretty general, they make sense in the context of the verb.

```
Verb: 되
 Occurrences: 2017
 Top Subjects: [('것+이', 44), ('문제+가', 19), ('도움+이', 18), ('사람+이', 17), ('대상+이', 16)]
  Centroid: 48
  Nearest Neighbors:
   것: 0.6285
   내용: 0.5977
   결과물: 0.5974
   방법: 0.5892
   주제: 0.5671
 Top Objects: [('아이+들+을', 1), ('화살표+를', 1), ('제공자+들+을', 1)]
  Centroid: 3
  Nearest Neighbors:
   아이: 0.8817
   캣: 0.4573
   님프: 0.4513
   베이비: 0.4408
   마우스: 0.4403
 Top Modifiers: [('안', 91), ('것+이+다', 70), ('수', 49), ('잘', 17), ('것+이+ㅂ니다', 17)]
  Centroid: 50
  Nearest Neighbors:
   것: 0.8181
   수: 0.6364
   경우: 0.6096
   만큼: 0.5828
   가능성: 0.5722
 Top Before: [('안', 92), ('있게', 48), ('알게', 42), ('것이', 31), ('갖게', 28)]
  Centroid: 5
  Nearest Neighbors:
   잘: 0.7229
   오래: 0.7068
   많이: 0.5278
   안: 0.5064
   이: 0.5016
 Top After: [('.', 917), ('것이다', 179), ('것은', 50), ('수', 49), ('있다', 40)]
  Centroid: 16
  Nearest Neighbors:
   것: 0.7026
   때: 0.6275
   수: 0.6233
   거: 0.6019
   뿐: 0.5659
```



We can further demonstrate the idea that the less common a word is, the more clustered it tends to be. With the word 변하 (*pyen.ha.*, “to change (intrans.)”), we get a much more clustered view.

We wanted to demonstrate this concept with our 20-40% frequency group, but we found that we were unable to compute a good graphical representation with verbs that have a much lower frequency (those verbs have about 7 occurrences in our data set).

Looking at more information about the verb, we again see that the neighbors tend to me more specific to the verb, as we get neighbors that mean things like “location”, “material”, “shape”, all words that are relevant to the verb “change”.

```
Verb: 변하 (pyen.ha., “to change”)
Computed verb sets on-the-fly for '변하' (normalized: '변하')
 Occurrences: 51
 Top Subjects: [('사물+이' “work, product”, 1), ('집+이' “house”, 1), ('세상+이' “the world”, 1), ('언어+가' “word”, 1), ('구조+가' “structure”, 1)]
  Centroid in_vocab (computed): 19
 Top Objects: [('행동+을' “behaviour”, 1), ('위치+를' “location”, 1)]
  Centroid in_vocab (computed): 2
 Top Modifiers: [('것+이+다' “thing”, 5), ('수' “ability”, 3), ('오+았+다' “has come”, 2), ('난장판+으로' “cacophonously”, 1), ('소녀+로' “girl”, 1)]
  Centroid in_vocab (computed): 47
 Top Before: [('신은' “worn", 2), ('크게' “largely”, 2), ('옮아가고' “spread”, 1), ('난장판으로', 1), ('소녀로', 1)]
  Centroid in_vocab (computed): 5
 Top After: [('.', 9), ('수', 3), ('것이다', 3), ('않은' “not”, 2), ('왔다' “came”, 2)]
  Centroid in_vocab (computed): 5
```

Semantic Verb Generalizations (Task 2a)

Referring back to the verb 되다 (*twoi.*, “to become”) from earlier, most of the words were pretty general, but from the objects ([('children+is', 1), ('arrow+is', 1), ('providers+is', 1)]) and before ([('in', 92), ('have', 48), ('know', 42), ('thing', 31), ('have', 28)]) sets, we got the most information (which holds true for most words we found) as opposed to after ([('.', 917), ('is', 179), ('is', 50), ('can', 49), ('is', 40)]), modifiers ([('thing+is+this', 5), ('number', 3), ('came+at+da', 2), ('message+with', 1), ('girl+with', 1)]), and subjects ([('thing+this', 44), ('problem+is', 19), ('help+is', 18), ('person+this', 17)]).

```
Verb: 되
 Occurrences: 2017
 Top Subjects: [('것+이', 44), ('문제+가', 19), ('도움+이', 18), ('사람+이', 17), ('대상+이', 16)]
  Centroid: 48
  Nearest Neighbors:
   것: 0.6285 (“thing”)
   내용: 0.5977 (“contents, meaning, substance”)
   결과물: 0.5974 (“product, outcome, work”)
   방법: 0.5892 (“method”)
   주제: 0.5671 (“topic”)
 Top Objects: [('아이+들+을', 1), ('화살표+를', 1), ('제공자+들+을', 1)]
  Centroid: 3
  Nearest Neighbors:
   아이: 0.8817 (“child”)
   캣: 0.4573 (loanword from English “Cat”)
   님프: 0.4513 (loanword from English “Nymph”)
   베이비: 0.4408 (loanword from English “Baby”)
   마우스: 0.4403 (loanword from English “Mouse”)
 Top Modifiers: [('안', 91), ('것+이+다', 70), ('수', 49), ('잘', 17), ('것+이+ㅂ니다', 17)]
  Centroid: 50
  Nearest Neighbors:
   것: 0.8181 (“thing”)
   수: 0.6364 (“ability”)
   경우: 0.6096 (“circumstance, case, condition”)
   만큼: 0.5828 (“degree, amount, extent”)
   가능성: 0.5722 (“degree of possibility”)
 Top Before: [('안', 92), ('있게', 48), ('알게', 42), ('것이', 31), ('갖게', 28)]
  Centroid: 5
  Nearest Neighbors:
   잘: 0.7229 (“well”)
   오래: 0.7068 (“long time”)
   많이: 0.5278 (“many, a lot”)
   안: 0.5064 (“not”)
   이: 0.5016 (“this”; “be”)
 Top After: [('.', 917), ('것이다', 179), ('것은', 50), ('수', 49), ('있다', 40)]
  Centroid: 16
  Nearest Neighbors:
   것: 0.7026 (“thing”)
   때: 0.6275 (“instance, time”)
   수: 0.6233 (“ability”)
   거: 0.6019 (“that, thing”)
   뿐: 0.5659 (“just, only”)
```

Which Set is Best?

Objects

Pros

- Provides the clearest idea of what sort of semantic space a verb inhabits by what nouns it associates with

Cons

- Not applicable for intransitive verbs

Before

Pros

- Contains some objects
- Adverbs help distinguish intransitives
- Provides insight into common fossilized syntactic constructions

Cons

- Mostly the same set of generic adverbs (time, place, negation, etc)

Verb Generalizations (Task 2c)

Verb Generalizations: the Sequel

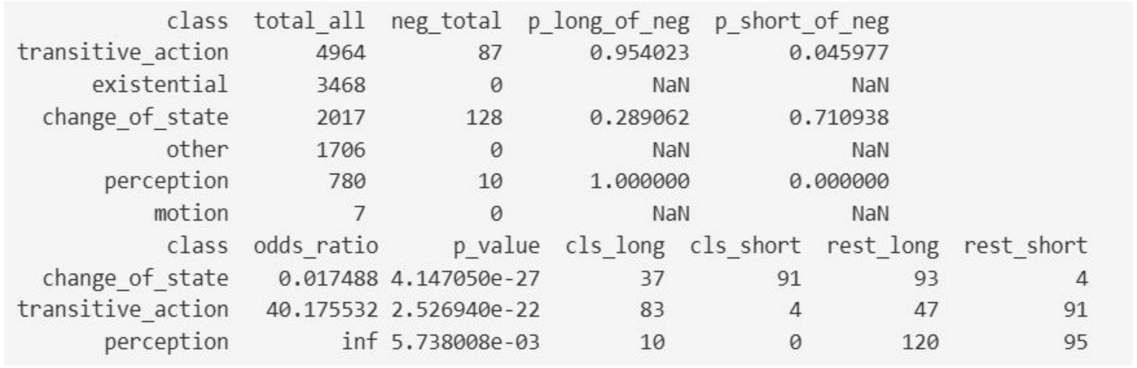
Korean features two primary methods by which verbs are negated.

- Method 1: Henceforth referred to as the Short Form. Constructed with 안 ("not") directly preceding the verb. Example: **안** 먹었어요 "I didn't eat"
- Method 2: Henceforth referred to as the Long Form. Constructed by suffixing the verb with -지 followed by the head verb 않다 ("to not"). Example: 먹**지 않**았어요 "I didn't eat"

Consideration
있다 (*iss.*, "to be, exist"; 3rd most common verb) and 이다 (*i.*, copula; most common verb) both have irregular negative verbs (없다 *ebs.* and 아니다 *a.ni.* respectively). The short form is never used with 있다 *iss.* or 이다 *i.*, but the long form is utilized.

We ultimately decided that the irregular suppletive negative forms were outside of the scope of our generalization and both verbs were omitted from our sample.

Analysis



```
             class  total_all  neg_total  p_long_of_neg  p_short_of_neg
transitive_action       4964         87       0.954023        0.045977
      existential       3468          0            NaN             NaN
  change_of_state       2017        128       0.289062        0.710938
            other       1706          0            NaN             NaN
       perception        780         10       1.000000        0.000000
           motion          7          0            NaN             NaN
             class  odds_ratio       p_value  cls_long  cls_short  rest_long  rest_short
  change_of_state    0.017488  4.147050e-27        37         91         93           4
transitive_action   40.175532  2.526940e-22        83          4         47          91
       perception         inf  5.738008e-03        10          0        120          95
```

We see here a strong preference for transitive verbs (those in the perception and transitive_action classes) to take the long negation and intransitive verbs (those in the change_of_state class) to take the short negation. The preference is only absolute for verbs of perception, however.

A possible explanation for this preference is that transitive verbs prefer to keep their objects near the verb, and there is a sort of contest of position between the object and the Short negative particle, which often pushes to negation into the Long Form so as to give both the object and negative particle maximal visibility/availability.

감사합니다!

"Thank you!"