Automatic Grammatical Case Prediction for Template Filling in Case-Marking Languages:

Implementation and Evaluation for Finnish

Johannes Laurmaa

The 8th Workshop on Research in Computational Linguistic Typology and Multilingual NLP (SIGTYP)

EACL 2026
RABAT • MOROCCO
Mars • March 24-29, 2026 • مارس

29.03.2026

01

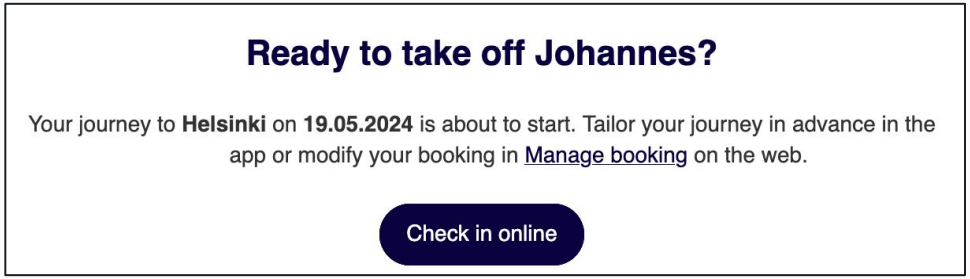
Motivation

Standard template filling is easy in English:

Helsinki

↓

🇬🇧 Your trip to [CITY] is starting



Example of a template-generated e-mail

01

Motivation

Template filling fails for highly inflected languages like Finnish:

The word must be inflected to the right case in order to be grammatically correct (`Helsinki` → `Helsinkiin`).

The Challenge: Grammatical case is context-dependent, making rule-based templates unfeasible

01

Motivation

Case	Example	Rough meaning
	Core cases	
Nominative	`talo`	house
Genitive	`talon`	of the house
Partitive	`taloa`	(some) house
Accusative	`talo(n)`	the house (object)
	Internal locative cases	
Inessive	`talossa`	in the house
Elative	`talosta`	out of the house
Illative	`taloon`	into the house
	External locative cases	
Adessive	`talolla`	at the house
Ablative	`talolta`	from the house
Allative	`talolle`	to the house
	Other cases	
Essive	`talona`	as a house
Translative	`taloksi`	into a house
Abessive	`talotta`	without a house
Instructive	`taloin`	by means of houses
Comitative	`taloineen`	with houses

Table 1: Finnish grammatical cases with the word `talo` (house) as an example.

Table 6: Number of examples per class in the evaluation dataset

Case	Number of examples	Percentage
Nom	75305	33.4%
Gen	54045	23.9%
Par	31256	13.9%
Ine	17407	7.7%
Ill	11045	4.9%
Ela	10907	4.8%
Ade	10284	4.6%
Ess	4910	2.2%
All	4885	2.2%
Abl	2132	0.9%
Tra	1498	0.7%
Ins	1203	0.5%
Abe	289	0.1%
Com	230	0.1%
Total	225396	100.0%

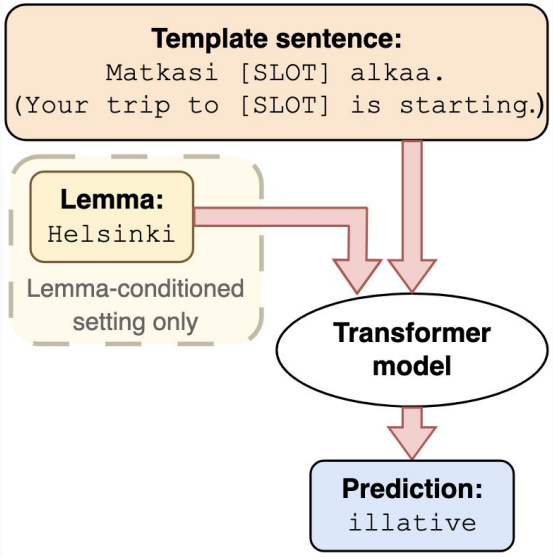
02

Our contributions

1. We formalize the case selection problem into two settings:
   - **Slot-only**: only the template sentence is provided
   - **Lemma-conditioned**: in addition of the template sentence, the noun is provided in its base form

2. We provide a recipe for the automatic construction of a dataset from raw text via morphological analysis.

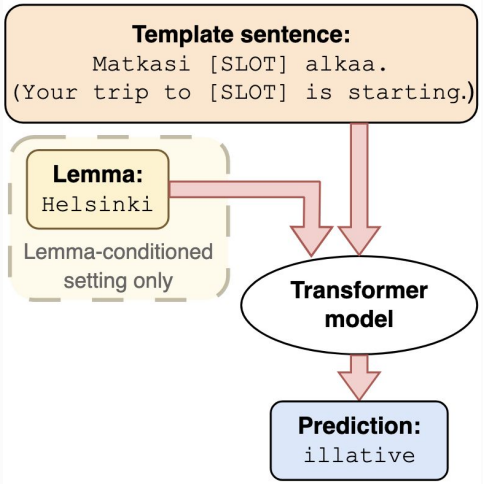
3. We train a transformer model to predict the grammatical case.

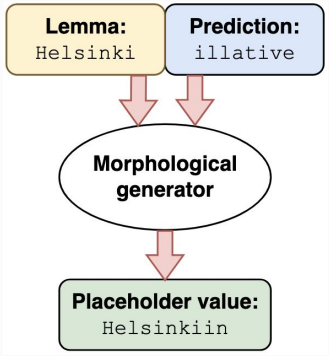


02

Our contributions

At inference time, you can use an existing rules-based morphological generator to inflect the word to the predicted grammatical case, before inserting it as the template placeholder.





03

Methodology – Dataset Creation

A dataset can be created from any raw text with the following recipe:

1. **Input**: Raw text
2. **Analysis**: Extract the lemma and case of nouns using a morphological analyzer
3. **Create samples**: Generate samples in slot-only and lemma-conditioned settings

`Helena rakentaa taloa.`

↓

Word	Lemma	POS	Case
`Helena`	`Helena`	noun	nominative
`rakentaa`	`/`	verb	/
`taloa`	`talo`	noun	partitive

↓

Input	Label	Setting
`Helena rakentaa [MASK].`	Partitive	Slot-only
`Helena rakentaa talo.`	Partitive	Lemma-conditioned

03

Methodology – Training

Models: we fine-tune two transformer models:

- XLM-RoBERTa (multilingual)
- FinBERT (Finnish)

Dataset: news from Yle (Finland's national public broadcasting company):

- 2019 news for training and validation
- 2021 news for testing

Methodology – Baselines & Uncertainty

Baselines: no clear baselines are readily available.
We compare our approach to two baselines:

- Majority class: always predict top class
- Adpositions: map pre- and post-positions around the noun to grammatical cases

Aleatoric uncertainty: multiple predictions can produce grammatically correct results.
The model does not know the real user intent.
In our previous example, although the first is more probable, these two outputs are technically correct:

🇬🇧 Your trip **to** Helsinki is starting

🇫🇮 Matkasi [CITY] alkaa

Helsinki**in**

🇬🇧 Your trip **from** Helsinki is starting

🇫🇮 Matkasi [CITY] alkaa

Helsingi**stä**

04

Results

	Lemma-conditioned		Slot-only		Baselines	
Metric	FinBERT	XLM-R	FinBERT	XLM-R	Majority class	Adpositions
Top-1						
Precision	89.1%	85.2%	73.3%	51.9%	2.5%	19.4%
Recall	81.1%	72.2%	63.0%	41.6%	7.1%	7.7%
F1 score	84.2%	75.3%	66.7%	44.7%	3.7%	5.0%
Accuracy	91.4%	87.8%	82.6%	80.8%	34.9%	35.7%
Top-3						
Recall	93.3%	90.8%	84.4%	73.9%	21.4%	21.7%
Accuracy	96.0%	94.1%	90.0%	89.6%	72.4%	72.6%

Table 2: Model performance in lemma-conditioned and slot-only settings. Baselines are slot-only heuristics. Precision, recall, and F1 are macro-averaged over the 14 classes.

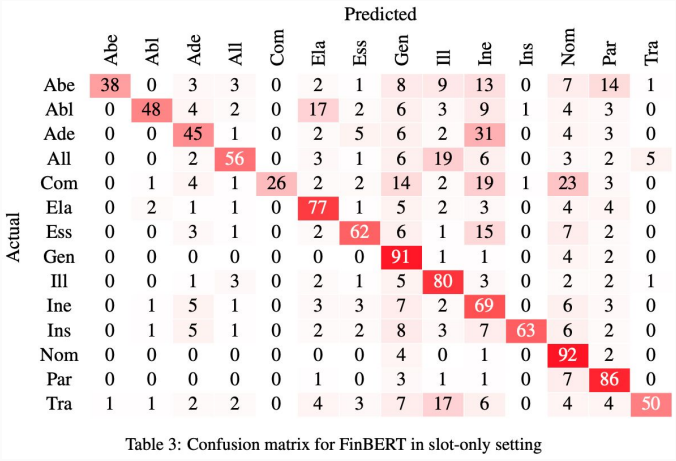
Table 5: Top-3 performance metrics by case (lemma-conditioned setting)

Case	Recall @ 3
Abe	97%
Abl	95%
Ade	93%
All	94%
Com	71%
Ela	95%
Ess	94%
Gen	96%
Ill	95%
Ine	93%
Ins	95%
Nom	98%
Par	96%
Tra	97%
Micro avg	96%
Macro avg	93%

Main results:

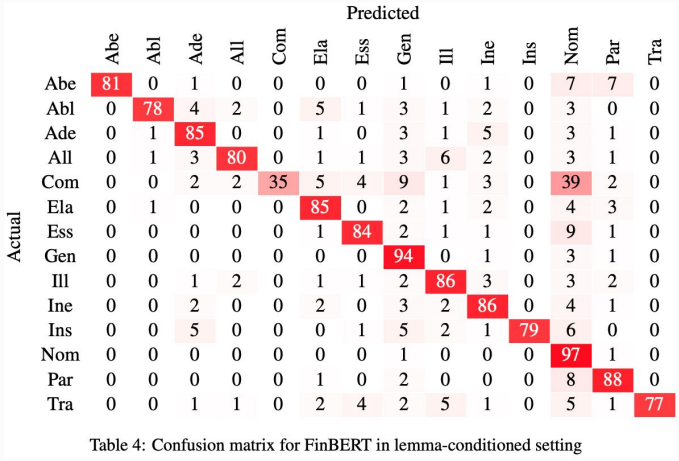
- The low baseline performance reflects the complexity of the task for rule-based methods
- Performance is higher when using a monolingual model
- Providing the lemma during inference helps significantly

Results - Slot-only vs. Lemma-conditioned



Actual \ Predicted	Abe	Abl	Ade	All	Com	Ela	Ess	Gen	Ill	Ine	Ins	Nom	Par	Tra
Abe	38	0	3	3	0	2	1	8	9	13	0	7	14	1
Abl	0	48	4	2	0	17	2	6	3	9	1	4	3	0
Ade	0	0	45	1	0	2	5	6	2	31	0	4	3	0
All	0	0	2	56	0	3	1	6	19	6	0	3	2	5
Com	0	1	4	1	26	2	2	14	2	19	1	23	3	0
Ela	0	2	1	1	0	77	1	5	2	3	0	4	4	0
Ess	0	0	3	1	0	2	62	6	1	15	0	7	2	0
Gen	0	0	0	0	0	0	0	91	1	1	0	4	2	0
Ill	0	0	1	3	0	2	1	5	80	3	0	2	2	1
Ine	0	1	5	1	0	3	3	7	2	69	0	6	3	0
Ins	0	1	5	1	0	2	2	8	3	7	63	6	2	0
Nom	0	0	0	0	0	0	0	4	0	1	0	92	2	0
Par	0	0	0	0	0	1	0	3	1	1	0	7	86	0
Tra	1	1	2	2	0	4	3	7	17	6	0	4	4	50

Table 3: Confusion matrix for FinBERT in slot-only setting



Actual \ Predicted	Abe	Abl	Ade	All	Com	Ela	Ess	Gen	Ill	Ine	Ins	Nom	Par	Tra
Abe	81	0	1	0	0	0	0	1	0	1	0	7	7	0
Abl	0	78	4	2	0	5	1	3	1	2	0	3	0	0
Ade	0	1	85	0	0	1	0	3	1	5	0	3	1	0
All	0	1	3	80	0	1	1	3	6	2	0	3	1	0
Com	0	0	2	2	35	5	4	9	1	3	0	39	2	0
Ela	0	1	0	0	0	85	0	2	1	2	0	4	3	0
Ess	0	0	0	0	0	1	84	2	1	1	0	9	1	0
Gen	0	0	0	0	0	0	0	94	0	1	0	3	1	0
Ill	0	0	1	2	0	1	1	2	86	3	0	3	2	0
Ine	0	0	2	0	0	2	0	3	2	86	0	4	1	0
Ins	0	0	5	0	0	0	1	5	2	1	79	6	0	0
Nom	0	0	0	0	0	0	0	1	0	0	0	97	1	0
Par	0	0	0	0	0	1	0	2	0	0	0	8	88	0
Tra	0	0	1	1	0	2	4	2	5	1	0	5	1	77

Table 4: Confusion matrix for FinBERT in lemma-conditioned setting

The lemma can give extra context to improve performance:

- Helps disambiguate between possible cases
  - Example: city names are often inflected to locative cases
- The case can sometimes depend on the noun itself
  - Example: in the same sentence, the city `Helsinki` would be inflected to internal locative cases, while the city `Tampere` would be inflected to external locative cases

05

What about LLMs?

LLMs are trained to form grammatically correct sentences. They can therefore be used for template-filling too.

However:

- Our experiment shows they do not necessarily achieve higher performance than Transformer models
- LLMs are heavier to run
- In template-filling scenarios, the flexibility of LLMs is not always desired
- Off-the-shelf LLMs often only produce a single truth, and not a probabilistic output

Prompt	Precision	Recall	F1	Accuracy
#1	48.2%	35.8%	34.9%	35.8%
#2	77.1%	71.6%	71.3%	71.9%
#3	81.1%	76.6%	76.7%	77.3%

Table 9: GPT-5 performance (macro averages) on the lemma-conditioned task.

```
Your task is to detect the matching
    grammatical case in Finnish
    sentences. You will be given a
    sentence with a placeholder. Your
    output should be grammatical case
    that best fits words that are
    inserted in the placeholder.
The possible cases are Nom, Gen, Par,
    Ine, Ela, Ill, Ade, Abl, All, Ess,
    Tra, Abe, Ins, Com.

# Example

Input: `Soitin eilen [kaveri].`
Output: `All`

# Task
Input: `{input_sentence}`
```

Figure 3: Prompt 1: Predict grammatical case name directly.

```
You will be given a sentence in Finnish.
     Your task is to inflect the word in
     the brackets to the correct
    grammatical case. Your output should
     contain the inflected word.

# Example
Input: `Soitin eilen [kaveri].`
Output: `kaverille`

# Task
Input: `{input_sentence}`
```

Figure 4: Prompt 2: Inflect word only.

```
You will be given a sentence in Finnish.
     Your task is to inflect the word in
     the brackets to the correct
    grammatical case. Your output should
     contain the corrected sentence.

# Example
Input: `Soitin eilen [kaveri].`
Output: `Soitin eilen kaverille.`

# Task
Input: `{input_sentence}`
```

Figure 5: Prompt 3: Rewrite whole sentence with word inflected.

06

Conclusion

Contributions

- We formalize the case selection problem for template-filling use cases
- We provide a recipe for the automatic construction of a dataset from raw text
- We trained transformer models predicting cases in Finnish

Limitations and next steps

- Aleatoric uncertainty
  ⇒ Define the upper bound on the achievable performance
- This work is currently restricted to Finnish news data
  ⇒ Apply the recipe on other languages and other domains
- In case multiple placeholders are present, our approach does not ensure global consistency
  ⇒ Multi-slot prediction

Any questions?

Johannes Laurmaa

johlaurmaa@gmail.com

LinkedIn: https://www.linkedin.com/in/jlaurmaa/

Code: https://github.com/joh17/case-detection/

Models: https://huggingface.co/johl