Yale

Revisiting the Effectiveness of Automatic N-Gram Rule Generation for Spelling Normalization in Filipino

Lj Flores[1] Dragomir Radev[1]

[1]LILY Lab, Yale University, New Haven, CT

LILY Lab

Motivation

Many NLP applications (e.g. Google Translate) can't understand or correct slang in Filipino

Our Contribution: We try a heuristic n-gram model, and show that it is (1) much better than augmented deep learning methods, and (2) computationally efficient and interpretable.

Dataset

- **Source:** 403 slang words from Meta comments
- **Annotation:** 3 Filipino volunteers, 398 examples, 83.8% inter-annotator agreement

Benchmarks

- **Language Models:** ByT5, Roberta-Tagalog
- **Semi-supervised Techniques:** Pi-Model (Π-Model), Autoencoding Augmentation (AE)
- **Baselines:** Google Translate correction function, DLD Only

N-Gram Model

- **Rule Generation:** Slide a window of length k over the word, and record w[i: i+k] → c[j: j+k] as a rule (Fig 1A); uses fact that many words are abbreviated by syllable (~1-2 letters)
- **Candidate Generation:** Recursively generate candidates by replacing each substring with all possible rules in the rule dictionary (Fig 1C)
- **Ranking Candidates:** Using (1) edit distance, or (2) Likelihood Score (See Fig 1B & 1C)

```
{'2l': ['tu', 'ka', 'tu', '2l', 'tu'],
 'lo': ['lo', 'lu', 'lo', 'lo'],
 'y':  ['y'],
 'ul': ['ul'],
 'lu': ['lu'],
 'uy': ['uy'],
 'n':  ['n', 'in', 'n', 'ya', 'na', 'ng',...]
```

```
 '2l': {'tu': 0.6, 'ka': 0.2, '2l': 0.2},
 'lo': {'lo': 0.75, 'lu': 0.25},
 'y':  {'y': 1.0},
 'ul': {'ul': 1.0},
 'lu': {'lu': 1.0},
 'uy': {'uy': 1.0},
 'n':  {'n': 0.708,   'in': 0.083, 'ya': 0.021,
        'na': 0.0625, 'ng': 0.020, 'on': 0.020},
 ...
```

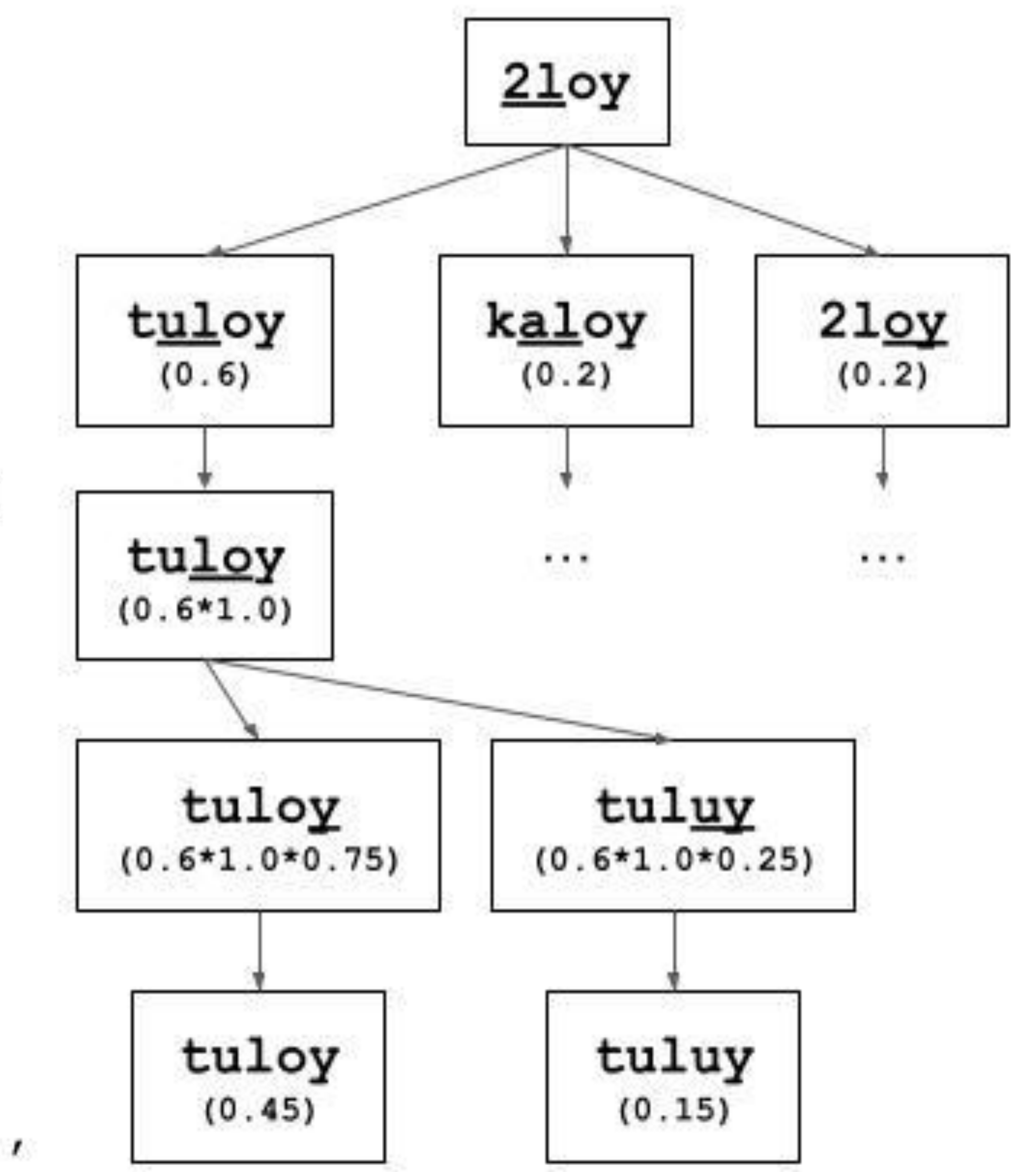

Fig 1. Candidate generation (left) and inference (right) example

Table 1. Performance of N-Gram Model and Benchmarks

Type	Model	Accuracy @ k (%) $k=1$	$k=3$	$k=5$	DLD Min	Mean	Max
N-Gram Based	N-Grams + DLD V1	**0.77**	**0.82**	**0.85**	**0.46**	2.91	4.73
	N-Grams + DLD V2	0.67	0.74	0.74	1.03	2.96	4.59
	N-Grams + Likelihood V1	0.17	0.38	0.58	1.22	3.50	5.29
	N-Grams + Likelihood V2	0.47	0.61	0.64	1.30	3.06	4.65
ByT5	Model Only	0.31	0.42	0.49	0.98	2.71	4.38
	Model + Π-Model	0.37	0.58	0.66	0.57	**2.06**	3.41
	Model + AE	0.04	0.04	0.04	4.28	6.69	10.2
Roberta-Tagalog	Model Only	0.00	0.00	0.00	5.79	15.3	56.7
	Model + Π-Model	0.00	0.00	0.00	5.69	16.5	69.2
	Model + AE	0.00	0.00	0.00	9.44	42.8	81.7
Baselines	DLD	0.45	0.67	0.72	0.59	2.28	**3.32**
	Google Translate	0.44	-	-	-	-	-

Results

- **N-Grams + DLD V1 has best accuracy;** +32% in accuracy @ 1 from the next best model (DLD)
- **Transparent model predictions allow for troubleshooting;** Errors when either (1) rule is not in the training set, or (2) similarity in spelling of the selected candidate to the actual candidate
- **N-Gram model trains in >1s on a CPU, performs inference in ~8.6ms,** in contrast LM with hyperparam tuning required ~6 GPU hours