Approaches to Pro-Drop in Machine Translation

Avani Pai and Cassie Maz

Summary

- What is a pro-drop language?
- Difficulties in with pro-drop in translation
- Importance to our Project
- Empty Categories (Chung and Gildea 2010)
- Model manipulation (Wang et al 2018)

What is Pro-Drop?

- Pro-Drop = pronoun dropping
- Omitting pronouns where they are inferable
- In some languages, pronoun still reflected in a verb's inflection
  - Example: Spanish (see table below)
    - "Yo hablo inglés" = "Habl**o** inglés" = "I speak English"
- In other languages, like Japanese, reflected pragmatically by larger context

I	hablo	We	hablamos
You (sg)	hablas	You (pl)	habláis
he/she/you (formal)	habla	they	hablan

Ex. 1（私は）フリーモントに住んでいます。

(watashi ha) furīmonto ni sunde imasu.

(I) live in Fremont.

Ex. 2（あなたが）どうして（私に）教えてくれなかったの？

（anata ga) doushite (watashi ni) oshiete kurenakatta no?

Why didn’t (you) tell (me)?

Challenges in Machine Translation

- “Translating these pro-drop languages into languages such as English where pronouns are regularly retained could be problematic because English pronouns have to be generated from nothing” (Chung & Gildea pg. 636)
- In languages like Chinese and Japanese, dropped pronouns are especially prevalent in dialogue and more casual speech. So MT models that are targeting such scenarios should be wary of this issue if they want high-quality translations

Relevance to our Project

- Our project investigates what a model is paying attention to when translating between pro-drop and non-pro-drop languages
- Japanese to English
  - Is the model recovering the pronouns?
  - If so, is it recovering the correct pronoun?
  - What could be affecting if/how pronouns are recovered?
- English to Japanese
  - Is the model dropping pronouns at similar rates to naturally spoken Japanese?
  - What could be affecting a small amount, accurate amount, or overinflated amount of pro-drop?
- Background Research: how has the issue of pro-drop been approached before?

Article 1: Effects of Empty Categories (Chung and Gildea 2010)

- Empty Categories: “Elements in parse trees that lack corresponding overt surface forms” (pg. 636)
- Previous Approaches:
  - NULL elements in word alignment (Brown et al 1993)
  - Phrase-based machine translation (used here)
- Part 1: Train Machine Translation Model on data w/ manually inserted ECs
- Part 2: Automatically inserting ECs in large data sets

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 1: Train model on data w/ manually inserted ECs
- Data:
  - Chinese Treebank (~5k sentences)
    - Pro-Drop: ~1 in 4 sentences
  - Korean Treebank (~5k sentences)
    - Pro-Drop: ~9 in 10 sentences
  - Annotated with null elements
- Training using Moses w/ default parameters
- Experimentation: leaving in/out different ECs to see how they affect results

Korean		
T	0.47	trace of movement
(NP *pro*)	0.88	dropped subject or object
(WHNP *op*)	0.40	empty operator in relative constructions
?	0.006	verb deletion, VP ellipsis, and others

Chinese		
(XP (-NONE- *T*))	0.54	trace of A'-movement
(NP (-NONE- *))	0.003	trace of A-movement
(NP (-NONE- *pro*))	0.27	dropped subject or object
(NP (-NONE- *PRO*))	0.31	control structures
(WHNP (-NONE- *OP*))	0.53	empty operator in relative constructions
(XP (-NONE- *RNR*))	0.026	right node raising
(XP (-NONE- *?*))	0	others

Table 1: List of empty categories in the Korean Treebank (top) and the Chinese Treebank (bottom) and their per-sentence frequencies in the training data of initial experiments.

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 1: Train model on data w/ manually inserted ECs
- Results:
  - Translating Chinese to English, best results when marking ALL null elements in training
  - Translating Korean to English, best results when marking dropped subjects/objects
- Analysis:
  - Certain ECs may be better at improving overall translations than others

		BLEU
Chi-Eng	No null elements	19.31
	w/ *pro*	19.68
	w/ *PRO*	19.54
	w/ *pro* and *PRO*	20.20
	w/ all null elements	**20.48**
Kor-Eng	No null elements	20.10
	w/ *pro*	**20.37**
	w/ all null elements	19.71

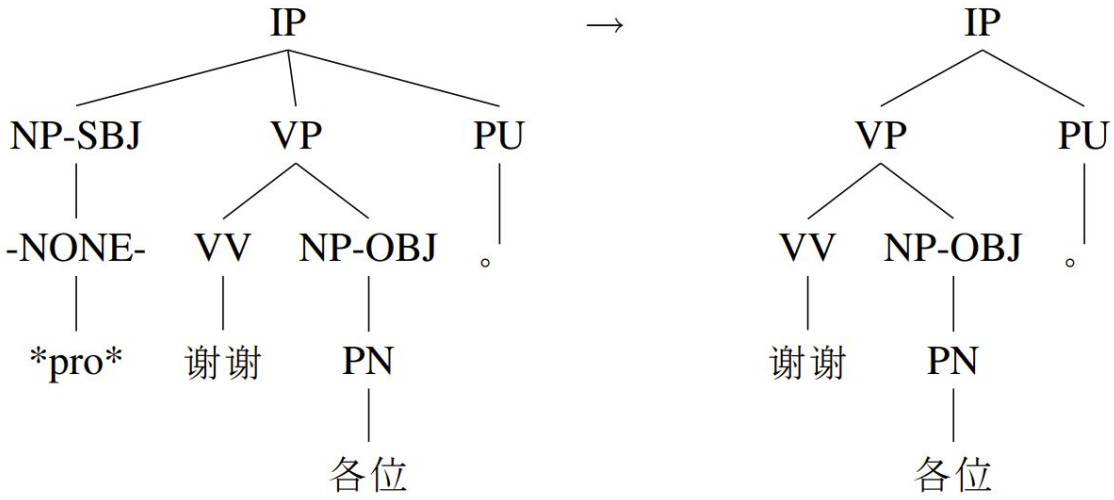
Table 2: BLEU score result of initial experiments. Each experiment has different empty categories added in. *PRO* stands for the empty category used to mark control structures and *pro* indicates dropped pronouns for both Chinese and Korean.

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Basic Idea: Remove annotations from data marked w/ ECs, and see how accurately annotations can be relabeled
- Methods for restoring ECs
  - Pattern Matching (Johnson 2002)
  - Conditional Random Fields
  - Parsing
- Data:
  - Chinese-English Treebank (~28K sentences)
  - Interested in recovering dropped pronouns (*pro*) and control structure markers (*PRO*)

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Method 1: Pattern Matching
  - Modeled after Johnson (2002)
    - Used tree fragment of empty node and all nodes co-indexed with it



(IP (NP-SBJ (-NONE- *pro*)) VP PU)

(IP VP PU)

Figure 1: An example of a tree with an empty node (left), the tree stripped of an empty node (right), and a pattern that matches the example. Sentences are parsed without empty nodes and if a tree fragment (IP VP PU) is encountered in a parse tree, the empty node may be inserted according to the learned pattern (IP (NP-SBJ (-NONE- *pro*)) VP PU).

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Method 1: Pattern Matching
  - Modeled after Johnson (2002)
    - Used tree fragment of empty node and all nodes co-indexed with it
  - Pattern 1 Modifications: Include parents and siblings of tree fragment
    - Necessary to disambiguate *pro* vs *PRO* matches
  - Pattern 2 Modifications: Like pattern 1, but include terminal information
    - (VP VV (IP (NP (-NONE- *PRO*)) VP))
    - (VP (VV 决 定) (IP (NP (-NONE- *PRO*)) VP))

PRO	
Count	Pattern
12269	(IP (NP (-NONE- *PRO*)) VP)
102	(IP PU (NP (-NONE- *PRO*)) VP PU)
14	(IP (NP (-NONE- *PRO*)) VP PRN)
13	(IP NP (NP (-NONE- *PRO*)) VP)
12	(CP (NP (-NONE- *PRO*)) CP)

pro	
Count	Pattern
10073	(IP (NP (-NONE- *pro*)) VP)
657	(IP (NP (-NONE- *pro*)) VP PU)
415	(IP ADVP (NP (-NONE- *pro*)) VP)
322	(IP NP (NP (-NONE- *pro*)) VP)
164	(IP PP PU (NP (-NONE- *pro*)) VP)

PRO	
Count	Pattern
2991	(VP VV NP (IP (NP (-NONE- *PRO*)) VP))
2955	(VP VV (IP (NP (-NONE- *PRO*)) VP))
850	(CP (IP (NP (-NONE- *PRO*)) VP) DEC)
765	(PP P (IP (NP (-NONE- *PRO*)) VP))
654	(LCP (IP (NP (-NONE- *PRO*)) VP) LC)

pro	
Count	Pattern
1782	(CP (IP (NP (-NONE- *pro*)) VP) DEC)
1007	(VP VV (IP (NP (-NONE- *pro*)) VP))
702	(LCP (IP (NP (-NONE- *pro*)) VP) LC)
684	(IP IP PU (IP (NP (-NONE- *pro*)) VP) PU)
654	(TOP (IP (NP (-NONE- *pro*)) VP PU))

Table 5: Top five minimally connected patterns that match *pro* and *PRO* (top). Patterns that match both *pro* and *PRO* are shaded with the same color. The table on the bottom show more refined patterns that are given added context by including the parent and siblings to minimally connected patterns. Many patterns still match both *pro* and *PRO* but there is a lesser degree of overlap.

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Method 1: Pattern Matching
  - Modeled after Johnson (2002)
    - Used tree fragment of empty node and all nodes co-indexed with it
  - Pattern 1 Modifications: Include parents and siblings of tree fragment
    - Necessary to disambiguate *pro* vs *PRO* matches
  - Pattern 2 Modifications: Like pattern 1, but include terminal information
    - (VP VV (IP (NP (-NONE- *PRO*)) VP))
    - (VP (VV 决 定) (IP (NP (-NONE- *PRO*)) VP))
  - Pruned data of low-occurrence patterns

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Method 2: Conditional Random Field
  - Lafferty et al., 2001
  - 3 options for each word boundary:
    - Insert *pro*
    - Insert *PRO*
    - Leave as is
  - Model 1: words as features
  - Model 2: added POS tags as features
  - Model 3: added POS tags and parent node as features

Article 1: Effects of Empty Categories (Chung and Gildea)

- Part 2: Automatically inserting ECs in large data sets
- Method 3: Parsing
  - 1) Annotate Nonterminals (LHS) symbols with EC info, removed EC nodes
  - 2) Create CFG from annotated trees
  - 3) Used latent annotation learning procedures of Petrov et al. (2006) to modify CFG
  - 3) Used CFG to parse test sentences
- Discussion Q: The article mentions they first tried to use the unmodified CFG to parse test sentences, stating “this approach did not work well.” Thoughts as to why this might be?

Article 1: Effects of Empty Categories (Chung and Gildea)

Results: Recovering ECs

- Relatively low Precision, Recall, and F1 when compared to previous work done on English
- Takeaways:
  - Knowledge of tree structure important for *PRO* recovery
  - Local contexts/POS and/or machine learning important to *pro* recovery
- Big Question: When applying the best version of each recovery method, how is machine translation affected?

	PRO			*pro*		
	Prec.	Rec.	F1	Prec	Rec.	F1
Pattern 1	0.65	**0.61**	**0.63**	0.41	0.23	0.29
Pattern 2	0.67	0.58	0.62	0.46	0.24	0.31
CRF 1	0.66	0.31	0.43	0.53	0.24	0.33
CRF 2	**0.68**	0.46	0.55	**0.58**	0.35	**0.44**
CRF 3	0.63	0.47	0.54	0.54	0.36	0.43
Parsing	0.60	0.53	0.56	0.46	**0.39**	0.42

Table 7: Result of recovering empty nodes

Article 1: Effects of Empty Categories (Chung and Gildea)

Set-Up

- FBIS newswire data (~60K sentences)
- Moses w/ default parameters

Results:

“The machine translation system that used training data from the method that was overall the best in predicting empty elements performed the best.”

	BLEU	BP	*PRO*	*pro*
Baseline	23.73	1.000		
Pattern	23.99	0.998	0.62	0.31
CRF	**24.69***	1.000	0.55	0.44
Parsing	23.99	1.000	0.56	0.42

Table 8: Final BLEU score result. The asterisk indicates statistical significance at $p < 0.05$ with 1000 iterations of paired bootstrap resampling. BP stands for the brevity penalty in BLEU. F1 scores for recovering empty categories are repeated here for comparison.

Article 1: Effects of Empty Categories (Chung and Gildea)

Results: Training w/ recovered ECs

- “We show that even when automatic prediction of null elements is not highly accurate, it nevertheless improves the end translation result.”

A brief aside: Combining Recovery Methods:

- Chung and Gildea tried combining methods: CRF for *pro* and pattern matching for *PRO*.
- BLEU score of 24.24, lower than CRF alone
- What explanation did they give for this result?

Article 1: Effects of Empty Categories (Chung and Gildea)

Conclusion/Discussion

- More complicated methods for EC recovery: Gabbard et al. (2006)
- “We can also consider simpler methods where different algorithms are used for recovering different empty elements, in which case, we need to be careful about how recovering different empty elements could interact with each other” (pg. 644)
- “preprocessing the corpus to address a certain problem in machine translation is less principled than tackling the problem head on by integrating it into the machine translation system itself.” (pg. 644)

Translating Pro-Drop Languages with Reconstruction Models

Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, Qun Liu

Question & Approach

How can we improve neural machine translation (NMT) between a pro-drop language and a non-pro-drop language?

Option 1: Annotate the data with dropped pronoun (DP) information

Option 2: Modify the NMT architecture to somehow embed DP information

… Let’s do both!

Reconstruction for the win!

- Basic idea: recreate the input
- In this paper, the goal is to recreate a labelled version of the source sentence
  - embeds DP information
  - Minimize reconstruction loss to encourage “good” DP embeddings
- This objective is treated as an auxiliary task to the main task of translation

Input	等 我 搬进 来 (我) 可以 买 一台 泡泡机 吗 ?
Ref.	When I move in, can I get a bubble machine?

Reconstructor’s Inner Workings

1. The labelled source sentence & hidden states are fed into the reconstructor
2. Attempts to reconstruct the labelled sentence
3. Reconstructor outputs reconstruction score
    a. This will be used to measure how well the DPs were recalled from the hidden state representations
4. The reconstruction score is then linearly interpolated with the encoder/decoder scores to provide an overall translation score

Math for no reason?

$$J(\theta, \gamma, \psi) = \arg\max_{\theta,\gamma,\psi} \sum_{n=1}^{N} \Big\{ \underbrace{\log P(\mathbf{y}^n|\mathbf{x}^n;\theta)}_{likelihood}$$

$$+ \underbrace{\log R_{enc}(\hat{\mathbf{x}}^n|\mathbf{h}^n;\theta,\gamma)}_{enc\text{-}rec} + \underbrace{\log R_{dec}(\hat{\mathbf{x}}^n|\mathbf{s}^n;\theta,\psi)}_{dec\text{-}rec} \Big\} \qquad (7)$$

Experiment Setup

1. Prepare data
    a. Source sentence (x), labelled source sentence (x’), target sentence (y)
2. Train & Test several models
    a. Baseline EncDec (Encoder-Decoder)
    b. Baseline + DP annotated data
    c. Enc(+Recon)Dec
    d. EncDec(+Recon)
    e. Enc(+Recon)Dec(+Recon)

Baseline Model

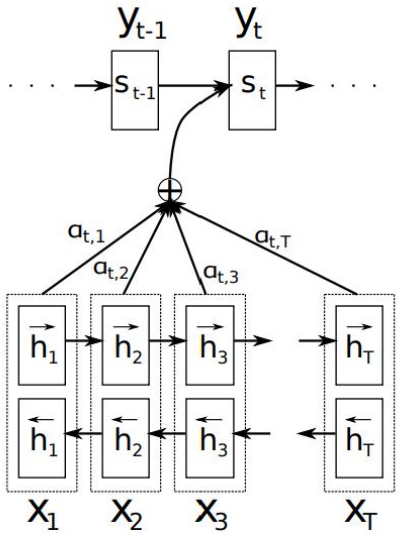


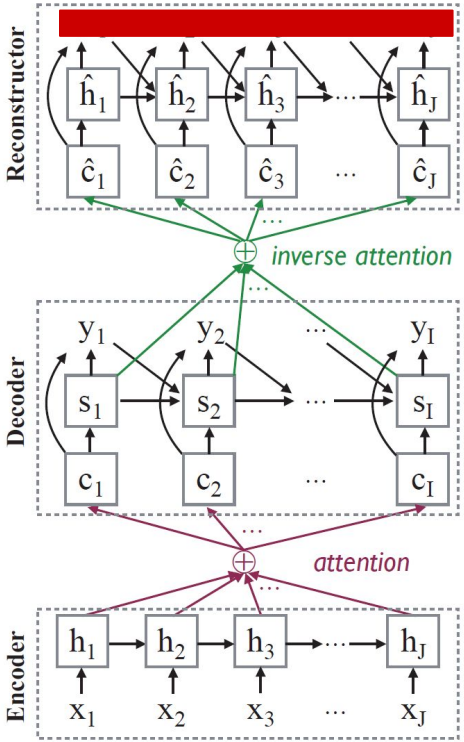
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Decoder

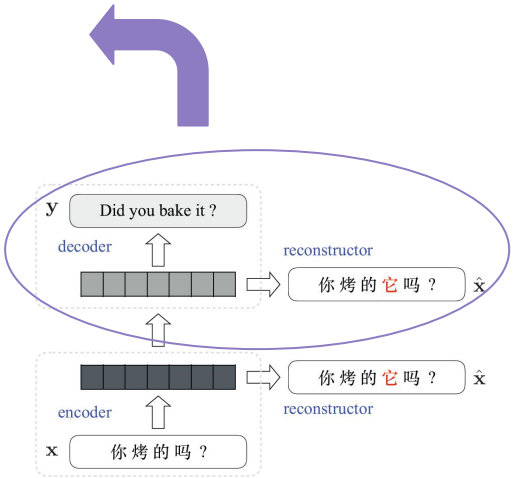
Encoder

Bahdanau, Cho, and Bengio 2015

Encoder+(Decoder+Reconstructor)



$x_1' x_2' x_3' \dots x_j'$



Tu et. al 2017

Data

- ~2M sentences of crawled data from a Chinese subtitle website
  - Randomly selected 2 episodes for tuning and another 2 for testing, rest is used as training
- DP annotation leverages previous work
  - Uses alignment information for training data
  - Uses a monologinual DP annotator model for testing data

Data	$\|S\|$	$\|W\|$		$\|P\|$		$\|V\|$		$\|L\|$	
		Zh	En	Zh	En	Zh	En	Zh	En
Train	2.15M	12.1M	16.6M	1.66M	2.26M	151K	90.8K	5.63	7.71
Tune	1.09K	6.67K	9.25K	0.76K	1.03K	1.74K	1.35K	6.14	8.52
Test	1.15K	6.71K	9.49K	0.77K	0.96K	1.79K	1.39K	5.82	8.23

Table 4: Number of sentences ($|S|$), words ($|W|$), pronouns ($|P|$), vocabulary ($|V|$), and averaged sentence length ($|L|$) comprising the training, tuning and test corpora. K stands for thousands and M for millions.

Results - ZhEn

Model	#Params	Speed		BLEU	
		Training	Decoding	Test	△
Baseline	86.7M	1.60K	2.61	31.80	– / –
Baseline (+DPs)	86.7M	1.59K	2.63	32.67†	+0.87 / –
+ enc-rec	+39.7M	0.71K	2.63	33.67†‡	+1.87 / +1.00
+ dec-rec	+34.1M	0.84K	2.18	33.48†‡	+1.68 / +0.81
+ enc-rec + dec-rec	+73.8M	0.57K	2.16	**35.08**†‡	**+3.28 / +2.41**
Multi-Source (Zoph and Knight 2016)	+20.7M	1.17K	1.27	32.81†	+1.01 / +0.14
Multi-Layer (Wu et al. 2016)	+75.1M	0.61K	2.42	33.36†	+1.56 / +0.69
Baseline (+DPs) + Enlarged Hidden Layer	+86.6M	0.68K	2.51	32.00†	+0.20 / -0.67

Table 5: Evaluation of translation performance for Chinese–English. "Baseline" is trained and evaluated on the original data, while "Baseline (+DPs)" is trained on the data labelled with DPs. "enc-rec" indicates encoder-side reconstructor and "dec-rec" denotes decoder-side reconstructor. Training speed is measured in words/second and decoding speed is measured in sentences/second with beam size being 10. The two numbers in the "△" column denote performance improvements over "Baseline" and "Baseline (+DPs)", respectively. "†" and "‡" indicate statistically significant difference ($p < 0.01$) from "Baseline" and "Baseline (+DPs)", respectively. All listed models except "Baseline" exploit the labelled source sentences.

Takeaways

Model	#Params	Speed		BLEU	
		Training	Decoding	Test	△
Baseline	86.7M	1.60K	2.61	31.80	– / –
Baseline (+DPs)	86.7M	1.59K	2.63	32.67†	+0.87 / –
+ enc-rec	+39.7M	0.71K	2.63	33.67†‡	+1.87 / +1.00
+ dec-rec	+34.1M	0.84K	2.18	33.48†‡	+1.68 / +0.81
+ enc-rec + dec-rec	+73.8M	0.57K	2.16	**35.08**†‡	**+3.28 / +2.41**

1. Modifying the model architecture along w/ providing DP info in the data is better than just having DP info
2. The encoder reconstructor & decoder reconstructor are encoding different patterns

Takeaways

Model	#Params	Speed		BLEU	
		Training	Decoding	Test	△
Baseline	86.7M	1.60K	2.61	31.80	– / –
Baseline (+DPs)	86.7M	1.59K	2.63	32.67†	+0.87 / –
+ enc-rec	+39.7M	0.71K	2.63	33.67†‡	+1.87 / +1.00
+ dec-rec	+34.1M	0.84K	2.18	33.48†‡	+1.68 / +0.81
+ enc-rec + dec-rec	+73.8M	0.57K	2.16	**35.08**†‡	**+3.28 / +2.41**
Multi-Source (Zoph and Knight 2016)	+20.7M	1.17K	1.27	32.81†	+1.01 / +0.14
Multi-Layer (Wu et al. 2016)	+75.1M	0.61K	2.42	33.36†	+1.56 / +0.69
Baseline (+DPs) + Enlarged Hidden Layer	+86.6M	0.68K	2.51	32.00†	+0.20 / -0.67

1. Performance improvement of reconstructor models is not just because of the increase in the number of parameters

Results & Takeaways - JpEn

Model	Test	△
Baseline (+DPs)	20.55	–
+ enc-rec + dec-rec	21.84	+ 1.29

Table 6: Evaluation of translation performance for Japanese–English.

1. BLEU score improves for JpEn translation as well, so the reconstructor architecture is potentially universal

Analysis

Using reconstruction in training only still leads to improved translation quality w/o speed tradeoff

Model	Test	△
Baseline	31.80	– / –
Baseline (+DPs)	32.67	+0.87 / –
+ enc-rec	33.67	+1.87 / +1.00
+ dec-rec	33.15	+1.35 / +0.48
+ enc-rec + dec-rec	**34.02**	**+2.22 / +1.35**

Table 7: Translation results when *reconstruction is used in training only while not used in testing*.

Changing reconstruction objective to recreate *the original* sentence still had improvements over baseline, but DP reconstruction model had larger gains.

Model	Test	△
Baseline	31.80	– / –
Baseline (+DPs)	32.67	+0.87 / –
+ enc-rec	33.21	+1.41 / +0.54
+ dec-rec	33.08	+1.28 / +0.41
+ enc-rec + dec-rec	**33.25**	**+1.45 / +0.58**

Table 8: Translation results when hidden states are *reconstructed into the original source sentence* instead of the source sentence labelled with DPs.

Other approaches to pro-drop

- Taira, H.; Sudoh, K.; and Nagata, M. 2012. Zero pronoun resolution can improve the quality of J-E translation. In Proceedings of the 6th Workshop on Syntax, Semantics and Structure in Statistical Translation, 111–118.
- Wang, L.; Tu, Z.; Zhang, X.; Li, H.; Way, A.; and Liu, Q. 2016a. A novel approach for dropped pronoun translation. In NAACL 2016, 983–993.
- Wang, L.; Tu, Z.; Zhang, X.; Liu, S.; Li, H.; Way, A.; and Liu, Q. 2017b. A novel and robust approach for pro-drop language translation. Machine Translation 31(1):65–87.

Discussion

- What do you think these papers imply about building successful machine translation systems?
- Do you think their approach is universal? Why or why not?

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In ICLR 2015, 1–15.

Chung, T., and Gildea, D. 2010. Effects of empty categories on machine translation. In EMNLP 2010, 636–645.

Gabbard, R., Kulick, S., & Marcus, M. (2006, June). Fully parsing the penn treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference* (pp. 184-191).

Johnson, M. (2002, July). A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 136-143).

Petrov, D. (2006). Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (pp. 433–440). Association for Computational Linguistics

Tu, Z.; Liu, Y.; Shang, L.; Liu, X.; and Li, H. 2017b. Neural machine translation with reconstruction. In AAAI 2017,3097–3103.

Wang, L., Tu, Z., Shi, S., Zhang, T., Graham, Y., & Liu, Q. (2018, April). Translating pro-drop languages with reconstruction models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).