A Survey of Developments in NAT

Yu Bao
2020-12
Background

- For the inputs of $x = \{x_1, \cdots, x_n\}$ and its target $y = \{y_1, \cdots, y_m\}$, there are two factorization for $p(y|x)$:
  - Autoregressive Factorization
    \[
    p(y|x) = \prod_i p_\theta(y_i | x, y_{<i})
    \]
  - Non-Autoregressive Factorization
    \[
    p(y|x) = \prod_i p_\theta(y_i | x)
    \]
Statistics

Non-Autoregressive Transformer (NAT)

2018
- ICLR(1)
- ICML(1)
- EMNLP(3)

2019
- AAAI(2)
- ACL(3)
- EMNLP(3)
- NeurIPS(2)
- Arxiv(3)

2020
- ICLR(1)
- AAAI(3)
- ICML(5)
- ACL(5)
- EMNLP(2)
- COLING(1)
- NeurIPS(1)
- INTERSPEECH(1)
- Arxiv(6)

Total / Translation
=5 / 5
=13 / 12
=25 / 19

Paper Collected by https://github.com/kahne/NonAutoregGenProgress
NAT in 2018

ICLR(1) ICML(1)
EMNLP(3)
The NAT works in 2018 laid down the **Basic Architecture** of research.

The diagram illustrates the different approaches to NAT:
- **Latent Transformer** (Kaiser et al.): Auto-regressive in the latent spaces.
- **Semi-AT** (Wang et al.): Auto-regressive in the *horizontal* direction.
- **Iterative NAT** (Lee et al.): Auto-regressive in the *vertical* direction.

The **NAT** (Gu et al.) is purely non-auto-regressive, forming the core of the architecture.
Performance Analysis (2018)

Performance on the WMT14 EN-DE, report the gap between NAT and its AT counterpart.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT</td>
<td>-4.28</td>
<td>-3.82</td>
</tr>
<tr>
<td>LT</td>
<td>-1.00</td>
<td>/</td>
</tr>
<tr>
<td>Semi-AT*</td>
<td>-0.21</td>
<td>/</td>
</tr>
<tr>
<td>Iterative-NAT</td>
<td>-3.03</td>
<td>-3.02</td>
</tr>
<tr>
<td>Weaker-AT Baseline</td>
<td>23.45</td>
<td>27.02</td>
</tr>
</tbody>
</table>

- Semi-AT ~ AT in performance.
- Iterative-NAT do not performs well.
NAT in 2019

NeurIPS(2) ACL(3) AAAI(2)
EMNLP(3)
Arxiv(3)

- Purely NAT (5)
- Latent Transformer (4)
- Semi-AT (1)
- Iterative-NAT (2)
The purely NAT works in 2019 focused on the **Training Techniques** or the **Decoder Inputs**.

**Training:** regularize the NAT with AT hints

- **NAT-REG** (Wang et al.)
  - Regularize the Final Hidden

- **HINT-NAT** (Wei et al.)
  - Regularize the Attention and Hidden with AT

- **imitate-NAT** (Wei et al.)
  - Regularize the Hidden with DAT

- **Reinforce-NAT** (Shao et al.)
  - Training with Sequential Objective

**Enhance the decoder inputs with target-side information**

- **ENAT** (Guo et al.)
  - Replace Decoder Inputs by Phrase Table and Word Mapping
Latent Transformer in 2019

The Latent Transformer works in 2019 focused on **Extracting Target-side Linguistic Information** as the learning object.

**Syn-ST** (Akoury et al.)
- Predict the syntax label sequence as inputs

**Reorder-NAT** (Ran et al.)
- Predict the aligned source order form inputs

**PNAT** (Bao et al.)
- Predict the aligned target order while decoding

**Flowseq** (Ma et al.)
- Extract target-side information with flow

Katzen schlafen viel

- **autoregressive**
  - Cats > sleep > a > lot

- **non-autoregressive**
  - Cats sleep a lot

- **semi-autoregressive**
  - Cats sleep > a lot

Latent Transformer
- NP1 > VP3 > Cats sleep a lot

SynST (ours)
- **°** > **°** > **°** > Cats sleep a lot

**Flow-graph diagram**
- Pseudo: I want my friends thank
- Source: I want to thank my friends

**Reordering Module**

**Encoder Module**

**Output Probabilities**

**Target Decoder**

**Target Encoder**

**Source Encoder**

**Source Encodings**

**One Step of Flow**

**Prior Flow**

Yu Bao

A Survey of Recent Developments in NAT
Semi-AT in 2019: Incorporate the **Structure Prediction** for conditional dependencies modeling of NAT.

![Diagram showing Semi-AT in 2019: Incorporate CRF into NAT decoding]
Iterative NAT works in 2019 focused on Explicitly Target-side Language Modeling for refinements.
Performance Analysis (2019)

Performance on the WMT14 EN-DE, report the gap between NAT and its AT counterpart.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT</td>
<td>-4.28</td>
<td>-3.82</td>
</tr>
<tr>
<td>NAT-REG</td>
<td>-2.69</td>
<td>-2.39</td>
</tr>
<tr>
<td>HINT-NAT</td>
<td>-2.10</td>
<td>-1.67</td>
</tr>
<tr>
<td>imitate-NAT</td>
<td>-3.65</td>
<td>-3.97</td>
</tr>
<tr>
<td>Reinforce-NAT*</td>
<td>-2.02</td>
<td>-1.61</td>
</tr>
<tr>
<td>ENAT</td>
<td>-3.13</td>
<td>-5.19</td>
</tr>
</tbody>
</table>

**Purely NAT**

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT</td>
<td>-1.00</td>
<td>/</td>
</tr>
<tr>
<td>Syn-ST*</td>
<td>-6.13</td>
<td>-5.23</td>
</tr>
<tr>
<td>Flowseq</td>
<td>-1.85</td>
<td>-0.76</td>
</tr>
<tr>
<td>PNAT</td>
<td>-2.92</td>
<td>-2.17</td>
</tr>
<tr>
<td>Reorder-NAT</td>
<td>-1.78</td>
<td>-0.57</td>
</tr>
</tbody>
</table>

**Latent Transformer**

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative-NAT</td>
<td>-3.03</td>
<td>-3.02</td>
</tr>
<tr>
<td>LevT</td>
<td>0.11</td>
<td>/</td>
</tr>
<tr>
<td>CMLM</td>
<td>-0.83</td>
<td>-0.54</td>
</tr>
</tbody>
</table>

**Semi-AT**

- Purely NAT and Latent Transformer is closed to the AT.
- Semi-AT \sim AT
- Iterative-NAT \succ AT in sometimes
NAT in 2020
ICLR(1) NeurIPS(1) ICML(3) ACL(4)
AAAI(3) EMNLP(2) COLING(1)
Arxiv(4)

- Purely NAT (7)
- Latent Transformer (2)
- Semi-AT (1)
- Iterative-NAT (7)
Purely NAT in 2020

Advanced **Training Techniques** for NAT.

- **FCL-NAT (Guo et al.)**
- **TCL-NAT (Liu et al.)**
- **BoN-NAT (Shao et al.)**
- **ENGINE (Tu et al.)**
- **EM-NAT (Sun et al.)**

Training NAT by curriculum learning from AT or Semi-AT

Bag-of-Ngrams differences as training objective

Training NAT to minimize the energy of AT

EM update the AT output for the NAT

---

**Figure 2:** The calculation process of \( \text{BoN}_4(\text{"get up")} \). First calculate the probability of the bigram “get up” in each sub-area and then accumulate the probabilities.
Purely NAT in 2020 (cont.)

Except the training techniques, improving NAT from **Data Augmentation** or **Model Architecture**.

- **Purely-NAT**
  - **NAT+Mono-KD (Zhou et al.)**
    - Leverage large monolingual corpora while taking the KD techniques for training.
  - **LAVA NAT**
    - Look Around: predict the current words condition on predicted context.
    - Vocabulary Attention: i.e expectation embedding as inputs
Latent Transformer in 2020


Iterative refinements in the hybrid (y and z) spaces:
- Training inference network: p(z|y, x), p(z|x), p(y|z, x)
- Inference:
  - Init $z = p(z|x)$
  - $y = p(y|x, z)$
  - update $z$ with $p(z|y, x)$

Iterative refinements in the fully latent spaces:
- Train inference network to estimate the left method gradient of $p(y|z, x)$ for given $z$.
- Inference:
  - Init $z$.
  - estimate the gradient of $z$.
  - update $z$. 
Semi-AT in 2020: To tackle the multi-modality problem by switching the autoregressive way.

Semi-AT in 2020: To tackle the multi-modality problem by switching the autoregressive way.

RecoverSAT (Ran et al.)

From Parallel within a block into Parallel between blocks.

Final translation: there are lots of farmers doing this today

Post-process
Iterative-NAT in 2020: Focus on the improvements over the CMLMs.

Enhance CTC with latent alignment and iterative refinements.

Update Training Methods

SMART (Ghazvininejad et al.)
AXE-CMLM (Ghazvininejad et al.)
DisCo (Kasai et al.)
JM-NAT (Guo et al.)

Iterative-NAT

Imputer (Saharia et al.)

Enhance CMLM

CCAN (Ding et al.)
AB-Net (Guo et al.)
Enhance CMLM with advanced training techniques. (mimics the decoding behavior)
Enhance CMLM with advanced training techniques. (cont.)

New Training Methods

AXE-CMLM (Ghazvininejad et al.)
Incorporate latent alignments while training.

JM-NAT (Kasai et al.)
Add masked-LM to the encoder, and take the span masking for decoder.

<table>
<thead>
<tr>
<th>Target $Y$</th>
<th>$it$</th>
<th>tastes</th>
<th>pretty</th>
<th>good</th>
<th>though</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment $\alpha : Y \rightarrow P$</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Model Predictions $P$ (Top 5)</td>
<td>but</td>
<td>it</td>
<td>tastes</td>
<td>delicious</td>
<td>ε</td>
</tr>
<tr>
<td></td>
<td>however</td>
<td>ε</td>
<td>makes</td>
<td>good</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>ε</td>
<td>that</td>
<td>looks</td>
<td>tasty</td>
<td>,</td>
</tr>
<tr>
<td></td>
<td>for</td>
<td>this</td>
<td>taste</td>
<td>fine</td>
<td>so</td>
</tr>
<tr>
<td></td>
<td>and</td>
<td>for</td>
<td>feels</td>
<td>exquisite</td>
<td>though</td>
</tr>
</tbody>
</table>
Iterative-NAT in 2020 (cont.)

Context-Aware Cross Attention: Enhance CMLM with localness cross-attention.

Enhance CMLM

CCAN
(Ding et al.)

(a) Vanilla Non-autoregressive model

(b) Localness modeling in fixed window
Adapter-Bert Networks: Incorporate BERT into CMLMs.
Performance Analysis (2020)

Performance on the WMT14 EN-DE, report the gap between NAT and its AT counterpart.

### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT</td>
<td>-4.28</td>
<td>-3.82</td>
</tr>
<tr>
<td>NAT-REG</td>
<td>-2.69</td>
<td>-2.39</td>
</tr>
<tr>
<td>HINT-NAT</td>
<td>-2.10</td>
<td>-1.67</td>
</tr>
<tr>
<td>imitate-NAT</td>
<td>-3.65</td>
<td>-3.97</td>
</tr>
<tr>
<td>Reinforce-NAT</td>
<td>-2.02</td>
<td>-1.61</td>
</tr>
<tr>
<td>ENAT</td>
<td>-3.13</td>
<td>-5.19</td>
</tr>
<tr>
<td>BoN-NAT</td>
<td>-3.67</td>
<td>-3.86</td>
</tr>
<tr>
<td>FCL-NAT</td>
<td>-1.55</td>
<td>-1.79</td>
</tr>
<tr>
<td>EM-NAT</td>
<td>-1.63</td>
<td>-2.33</td>
</tr>
<tr>
<td>LAVA-NAT*</td>
<td>1.12</td>
<td>0.17</td>
</tr>
<tr>
<td>TCL-NAT</td>
<td>-1.93</td>
<td>-0.78</td>
</tr>
</tbody>
</table>

**Purely NAT**

### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT</td>
<td>-1.00</td>
<td>/</td>
</tr>
<tr>
<td>LV-NAR</td>
<td>-6.13</td>
<td>-5.23</td>
</tr>
<tr>
<td>Flowseq</td>
<td>-1.85</td>
<td>-0.76</td>
</tr>
<tr>
<td>PNAT</td>
<td>-2.92</td>
<td>-2.17</td>
</tr>
<tr>
<td>Reorder-NAT</td>
<td>-1.78</td>
<td>-0.57</td>
</tr>
<tr>
<td>LV-NAR*</td>
<td>-1.00</td>
<td>/</td>
</tr>
<tr>
<td>Latent Transformer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative-NAT</td>
<td>-3.03</td>
<td>-2.02</td>
</tr>
<tr>
<td>LevT</td>
<td>0.11</td>
<td>/</td>
</tr>
<tr>
<td>CMLM</td>
<td>-0.83</td>
<td>0.54</td>
</tr>
<tr>
<td>AXE-CMLM</td>
<td>-4.22</td>
<td>-3.40</td>
</tr>
<tr>
<td>DisCo</td>
<td>-1.30</td>
<td>-0.38</td>
</tr>
<tr>
<td>CCAN</td>
<td>-1.70</td>
<td>/</td>
</tr>
<tr>
<td>JM-NAT</td>
<td>-0.35</td>
<td>-0.45</td>
</tr>
<tr>
<td>Imputer</td>
<td>0.40</td>
<td>0.10</td>
</tr>
<tr>
<td>AB-Net</td>
<td>0.65</td>
<td>0.88</td>
</tr>
<tr>
<td>SMART</td>
<td>-0.10</td>
<td>-0.03</td>
</tr>
</tbody>
</table>

**Semi-AT**

<table>
<thead>
<tr>
<th>Methods</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-AT</td>
<td>-0.21</td>
<td>/</td>
</tr>
<tr>
<td>NAT-DCRF</td>
<td>-0.61</td>
<td>-1.25</td>
</tr>
<tr>
<td>Recover-nat</td>
<td>-0.06</td>
<td>-0.28</td>
</tr>
</tbody>
</table>

**Iterative-NAT**

- Except purely NAT, other genre narrow the gap into 1.0 BLEU.

求放过我吧

Transformer

- Semi-AT = AT
- Iterative-NAT > AT
- BERT makes NAT great.
Conclusions in NAT-Survey-2019

- Solved Problem:
  - Modeling dependencies is essential to NAT more than enhance the inputs.

- Used Techniques:
  - CRF or other imitated supervision: Morning flowers, collected in the evening.
  - Iterative refinements: Go south by driving the chariot north.

Conclusions

- A brief survey of NATs' work over the last three years' main conference paper or arxiv papers.

- As the analysis in the "Performance Analysis (20XX)"

  - NAT develops rapidly: The performance gap between the NAT and AT is minor now. (< 1.0 BLEU )

  - Nevertheless, narrowing the purely NAT and AT gap is still a challenge: Need more advanced training technique or new model architecture.

  - [Need background] The need to use a Transformer for knowledge distillation and reranking is still a shadow of the NAT works.

More details about the survey paper in: https://baoy-nlp.github.io/NAT-2020-survey.png
Q & A
Thanks!
Reference

Reference (cont.)

Reference (cont.)