Improved Lexically Constrained Decoding for Translation and Monolingual Rewriting

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, Benjamin Van Durme

Rewriting lots of sentences quickly, with constraints on the lexical items

We propose a new algorithm and show positive results when used to augment NLP tasks.

Improved Constrained Decoding

Better algorithm:
Multi-state Trie (MST)

Representing 3 negative constraints:
• “the small dog”
• “the small bird”
• “small cat”

Without tracking multiple states, “Here comes the small cat.” becomes a possible output!

Better algorithm:
Vectorized Dynamic Beam Allocation (VDBA) \[1\]

Our prior work, ParaBank (AAAI’19)\[2\]: sentential paraphrase generation via backtranslation with lexical constraints. Resulted in millions of Eng:Eng pairs, suitable for training rewriting systems

Data Augmentation via Rewriting

Given a rewriter and an NLP dataset you wish was larger than it was: perhaps generate various paraphrases of what you have?

Improved Constrained Decoding

Better algorithm:
Multi-state Trie (MST)

Better constraint placements (by BLEU)
Representing 3 negative constraints:
• “the small dog”
• “the small bird”
• “small cat”

None of the phrases should appear in output!

QA models benefit from more diverse training data.
We release pTREC-QA, expanded from TREC-QA\[4\].

Our prior work, ParaBank (AAAI’19)\[2\]: sentential paraphrase generation via backtranslation with lexical constraints. Resulted in millions of Eng:Eng pairs, suitable for training rewriting systems

Contributions

Better rewriting system (publically available) for lexically-constrained decoding that is about 600% faster with batching;
Demonstrated improvements in QA, low resource MT, and in MNLI on top of ELMo

Table 3: F1 scores on MNLI +Train denotes training on augmented data; +Arg denotes using a weighted aggregation. Scores on the development set are a weighted average between the matched (m) and mismatched (mm) portions of the dataset, while the test set scores are additionally broken down into each category.

Table 5: Experimental results on QA selection. TREC-QA\[4\] Answer Sentence Selection task of Wang et al. EMNLP ’07

QA models benefit from more diverse training data. We release pTREC-QA, expanded from TREC-QA\[4\].

Brittle NLI models can break under paraphrasing. We release pMNLI, a paraphrastic expansion of MNLI\[3\].

Low-resource MT benefits from augmentation for either side; En-Tr (12.4 ± 1.0 BLEU) Tr-En (15.6 ± 0.5 BLEU).

Contributions

Better rewriting system (publically available) for lexically-constrained decoding that is about 600% faster with batching;
Demonstrated improvements in QA, low resource MT, and in MNLI on top of ELMo

100% faster for positively constrained decoding; faster for positively constrained decoding with batching Better constraint placements (by BLEU)

Table 3: F1 scores on MNLI. +Train denotes training on augmented data; +Arg denotes using a weighted aggregation. Scores on the development set are a weighted average between the matched (m) and mismatched (mm) portions of the dataset, while the test set scores are additionally broken down into each category.

Table 5: Experimental results on QA selection. TREC-QA\[4\] Answer Sentence Selection task of Wang et al. EMNLP ’07

QA models benefit from more diverse training data. We release pTREC-QA, expanded from TREC-QA\[4\].

Brittle NLI models can break under paraphrasing. We release pMNLI, a paraphrastic expansion of MNLI\[3\].

Low-resource MT benefits from augmentation for either side; En-Tr (12.4 ± 1.0 BLEU) Tr-En (15.6 ± 0.5 BLEU).

Contributions

Better rewriting system (publically available) for lexically-constrained decoding that is about 600% faster with batching;
Demonstrated improvements in QA, low resource MT, and in MNLI on top of ELMo

Code is based on extensions to AWS Sockeye, an open source, enterprise NMT toolkit. Trained models and data are available at: http://nlp.jhu.edu/parabank