The Unreasonable Effectiveness of Word Embeddings for Social Media Text Processing

Xing Niu[1], Marine Carpuat[1], and Jimmy Lin[2]
[1]University of Maryland, College Park [2]University of Waterloo

Introduction

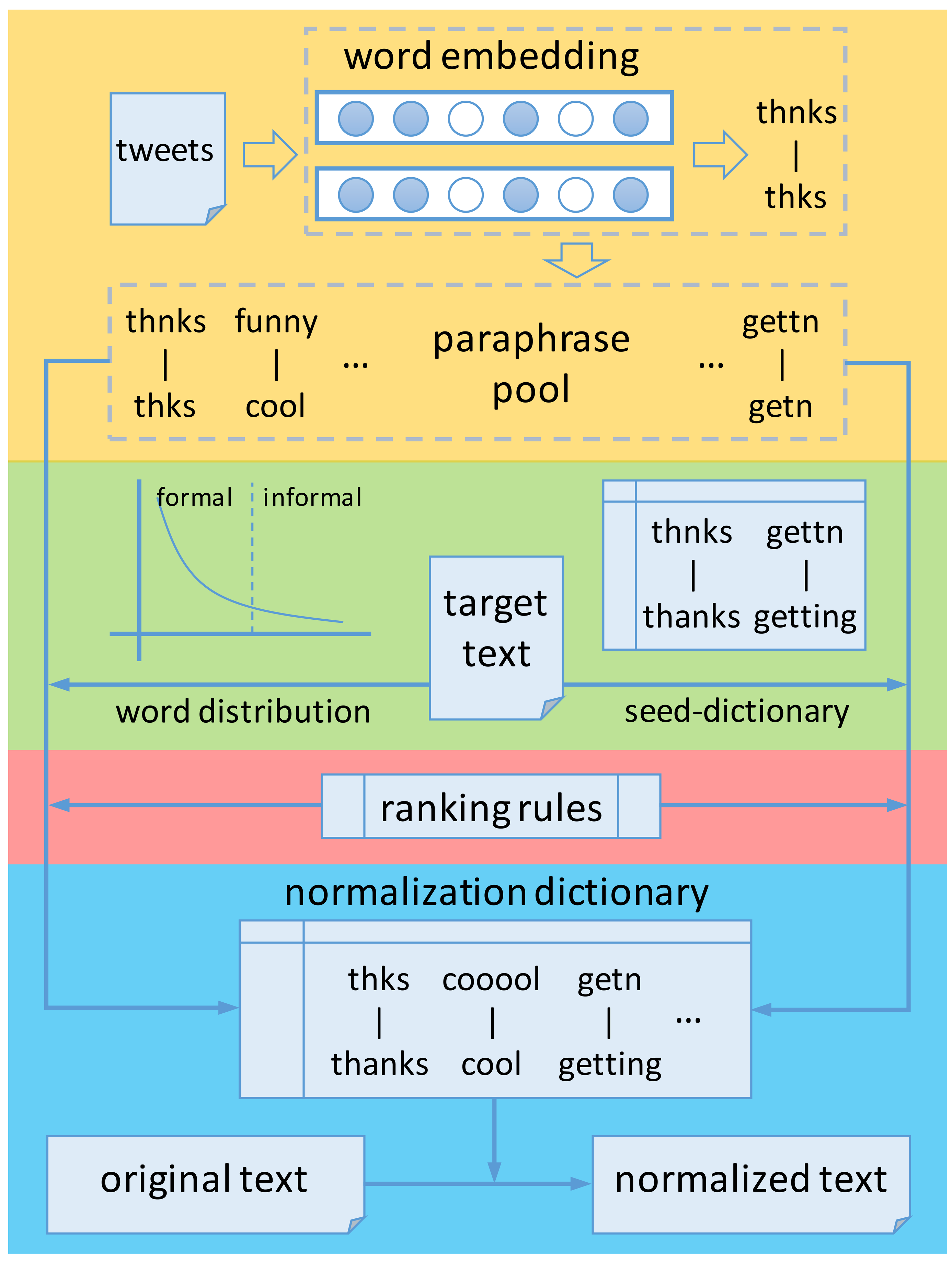
- Tweets, SMS, chats are challenging for Natural Language Processing.
- They are much more informal.
- **Text normalization** is one direction to address this issue.
- It makes informal text closer to traditional NLP corpora.
- For example:

Original tweet
@USER, r u cuming 2 MidCorner dis Sunday?
@USER, are you coming to MidCorner this Sunday?
Normalized tweet

- We propose an approach to lexical normalization that
  - ▷ requires no supervision
  - ▷ is exceedingly simple and flexible
- Results:
  - ▷ performs as well as off-the-shelf methods on lexical normalization
  - ▷ improves coverage and translation quality in a Weibo translation task

Our Approach

Workflow



1. **Training English word representations**
   - ▷ Skip-gram model from word2vec
   - ▷ A large monolingual corpus (e.g. Twitter)
   - ▷ Word pairs with high cosine similarity ⇒ paraphrase pool
2. **Generating normalization pairs**
   - ▷ Letting the target task inform what a standard normalized form should be.
   - ▷ Using the paraphrase pool to expand a given seed normalization dictionary.
   - ▷ Alternative: use word frequency information in representative normalized text to filter out paraphrases that are not normalization-related.
3. **Ranking normalization pairs**
   - ▷ Ranking by surface similarities: edit distance and character-level overlap/inclusion.
   - ▷ Building the normalization dictionary
4. **Text normalization**

Experimental Results

Training Data

Word embeddings

- Twitter 2013
- 88 million English tweets
- 1.1 billion tokens (875K distinct)

Lexical Normalization

Lexical normalisation for English tweets

- A shared task of ACL2015 Workshop on Noisy User-generated Text
- Task: normalizing non-standard words in English tweets to their canonical forms.
- Manually annotated data:
  - ▷ 2,950 training examples
  - ▷ 1,967 test examples
- Contrastive systems:

SAS-Ning The best system in the shared task.
- ▷ generates normalization candidates from the training data
- ▷ trains a binary classifier to select correct canonical form for a given token

UM UniMelb's normalization dictionary.
- ▷ is also built on Twitter corpus using customized distributional similarity
- ▷ requires a spell-checker, and annotated data for tuning parameters
- ▷ produces 41K normalization pairs

Method	Precision	Recall	F1
BASE	**0.9308**	0.7514	0.8315
IHS-RD	0.8469	**0.8083**	0.8272
SAS-Ning	0.9061	0.7865	0.8421
BASE+WE	0.9161	0.7800	0.8426
BASE+UM	0.8979	0.7938	0.8426
BASE+UM+WE	0.8842	**0.8083**	**0.8445**
ORACLE	0.9339	0.8188	0.8725
ORACLE+	0.9378	0.8858	0.9111

- Legend:

BASE The dictionary built on the training data.
IHS-RD The best unconstrained system in the shared task.
$$+WE Our Word Embeddings approach using $$ as the seed-dictionary.
ORACLE+ The dictionary built on the training+test data (theoretical upper-bound).
ORACLE Ruling out paraphrases in ORACLE+ but not in the pool (practical upper-bound).

- Our method performs as well as UM and SAS-Ning with fewer resources:
  - ▷ no supervision
  - ▷ no spell-checker
  - ▷ no complex feature engineering

Machine Translation

Training Data	Augmented Phrase-table?	BLEU	|OOV|
Weibo	×	14.78	2,203
Weibo	√	**15.03**	1,637
Weibo+BOLT	×	17.58	662
Weibo+BOLT	√	**17.64**	565

English-Chinese machine translation for social media text (Weibo)

- Data:
  - ▷ Weibo: 8,000 training, 1,250 dev and 1,250 test sentence pairs
  - ▷ BOLT: 1M out-of-domain sentence pairs (mix of formal and informal languages)
- Our method:
  - ▷ creating new phrase-table entries
  - ▷ by replacing formal source phrases (SP) with their unnormalized forms.
  - ▷ `SP ||| TP ||| f1 f2 f3 f4 ...`
- Results:
  - ▷ augmenting phrase-table helps coverage and BLEU most in low resource setting
  - ▷ but still helps translate some OOV in large data setting

xingniu@cs.umd.edu, marine@cs.umd.edu, jimmylin@uwaterloo.ca