From Paraphrase Database to Compositional Paraphrase Model and Back

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu
Transactions of the Association for Computational Linguistics, vol. 3, pp. 345-358, 2015.
(To be presented at EMNLP 2015)

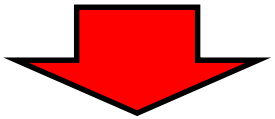
橋本和真
東京大学・鶴岡研究室
博士課程１年

この論文の特徴

- 二つの言い換えデータセットの構築
  - Annotated-PPDB
    - 言い換えデータベースPPDBの一部にアノテーション
  - ML-Paraphrase
    - バイグラムの意味的類似度を測るデータセットに再アノテーション
- 言い換えデータセットでの意味構成モデルの学習と評価
  - 先行研究の手法を上回る結果
- なぜこの論文か？
  - 今後の研究の役に立ちそうだから

Para**p**hrase **D**ata**b**ase: PPDB

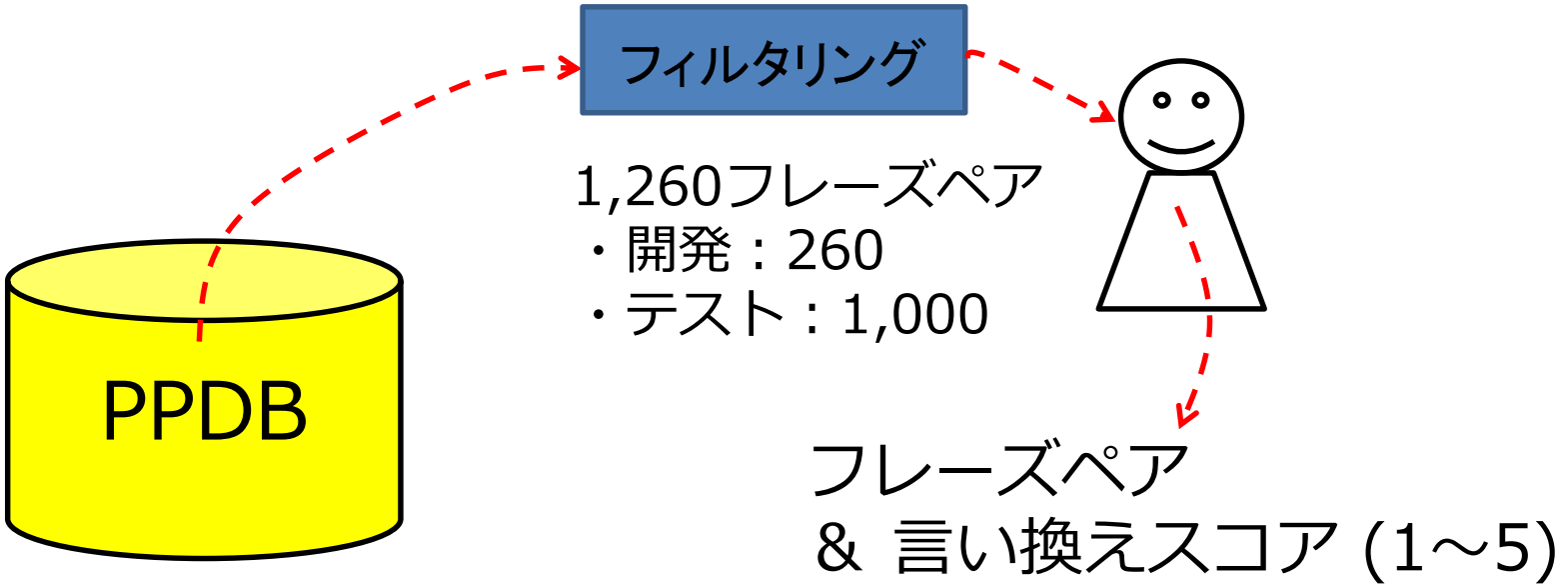
- 大規模なパラレルコーパスから自動構築された言い換えデータベース (Ganitkevitch et al., 2013)
  - 特定のフレーズがデータベースに無いと利用できない
  - 扱いたいフレーズの数だけパラメータが増加する
  - 自動構築なので必ずしも質が良いとは限らない



PPDBの一部にアノテーションして、
質の高い言い換えデータセットを構築
&
意味構成のモデルを学習・評価

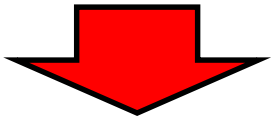
データセット1: Annotated-PPDB

- PPDBの一部を人手でアノテーション (具体的な手順は論文を参照)
  - 単語の重複度などでフィルタリング
  - フレーズの長さの制限
  - Amazon Mechanical Turkによるアノテーション



データセット2: ML-Paraphrase

- MLデータセット (Mitchell and Lapata, 2010)
  - 形容詞-名詞、動詞-目的語、名詞-名詞からなるバイグラムの意味的類似度をアノテーションしたもの
- このデータセットでの意味的類似度とは？
  - 同じ意味 (言い換え) だけでなく、トピック的な類似性
    - 例) television set と television programme
    - 例) older man と elderly woman



「言い換え」に着目した類似度スコアを著者の二人が改めて付与

言い換えモデル：意味構成関数

- Recursive Neural Networks (RNNs) (Socher et al., 2012)
  - フレーズを句構造解析 (二分木構造)
  - リーフノード (単語) は単語ベクトル
  - それ以外は子ノードのベクトルから再帰的に計算：

重み行列　　バイアス項

$$g(p) = f(W[g(c_1); g(c_2)] + b)$$

親ノードのベクトル　子ノードのベクトル

学習パラメータ：
RNNの重み行列とバイアス、単語ベクトル

言い換えモデル：目的関数と学習

- 言い換えデータベースを用いた言い換えモデルの学習
  - ミニバッチAdaGrad (Duchi et al., 2011) で学習

言い換えペアの
フレーズベクトルの内積

データベースに無いペアの
フレーズベクトルの内積

マージン

$$\min_{W,b,W_w} \frac{1}{|X|} \left(\sum_{\langle x_1, x_2 \rangle \in X} \max(0, \delta - g(x_1) \cdot g(x_2) + g(x_1) \cdot g(t_1)) + \max(0, \delta - g(x_1) \cdot g(x_2) + g(x_2) \cdot g(t_2)) \right) + \lambda_W (\|W\|^2 + \|b\|^2) + \lambda_{W_w} \|W_{w_{initial}} - W_w\|^2$$

事前学習した
word embeddingに基づく正則化

言い換えモデル：負例サンプリング

- ミニバッチ中に存在するフレーズの中で、対象フレーズと最も類似度が高いフレーズを負例として選択

$$\min_{W,b,W_w} \frac{1}{|X|} \Bigg(\sum_{\langle x_1, x_2 \rangle \in X} \max(0, \delta - g(x_1) \cdot g(x_2) + g(x_1) \cdot g(t_1)) + \max(0, \delta - g(x_1) \cdot g(x_2) + g(x_2) \cdot g(t_2)) \Bigg) + \lambda_W (\|W\|^2 + \|b\|^2) + \lambda_{W_w} \|W_{w_{initial}} - W_w\|^2$$

単語ベクトルの事前学習

- 全ての実験において、Skip-gram (Mikolov et al., 2013) を English Wikipediaで学習した単語ベクトルを使用
- Skip-gramベクトルの再学習
  - PPDBのXLサイズから抽出した単語レベルの言い換えデータを用いて再学習
    - PARAGRAMベクトル

- PARAGRAMベクトルで初期化して以降の意味構成関数の学習を行う

実験１：バイグラム言い換えの評価

- PPDBのXLサイズから学習データを抽出
  - 形容詞-名詞 (JN): 133,997ペア
  - 動詞-目的語 (VN): 62,640ペア
  - 名詞-名詞 (NN): 35,601ペア
- 単語・フレーズのベクトルの次元: 25
- 評価
  - MLとML-Paraphrase
    - 人手でフレーズペアにつけた意味的類似度・言い換え度合のスコアと、フレーズベクトルの類似度スコアのスピアマン相関係数

実験１：バイグラム言い換えの評価

- 自身の手法 (Hashimoto et al., 2014) との比較
  - MLでは同程度の性能
  - ML-ParaphraseではPARAGRAMが高性能
    - 言い換えタスクでは、言い換えの教師データが有効
    - Hashimoto et al. (2014) ではラベル無しコーパスのみ利用

Model			Mitchell and Lapata (2010) Bigrams				ML-Paraphrase			
word vectors	n	comp.	JN	NN	VN	Avg	JN	NN	VN	Avg
skip-gram	25	+	0.36	0.44	0.36	0.39	0.32	0.35	0.42	0.36
PARAGRAM	25	+	0.44*	0.34	0.48*	0.42	0.50*	0.29	**0.58***‡	0.46
PARAGRAM	25	RNN	**0.51***†	0.40†	**0.50***‡	**0.47**	**0.57***‡	**0.44**†	0.55*	**0.52**
Hashimoto et al. (2014)			0.49	0.45	0.46	**0.47**	0.38	0.39	0.45	0.41
Mitchell and Lapata (2010)			0.46	**0.49**	0.38	0.44	-	-	-	-
Human			-	-	-	-	0.87	0.64	0.73	0.75

実験２：Annotated-PPDBでの評価

- PPDB中のデータの質の判別をするタスク
  - PPDBで学習した言い換えモデルは、質の良い言い換えペアとそうでないペアを判別できるか？
    - Annotated-PPDBのスコアとの相関が高いか？
- PPDBのXLサイズから学習データを抽出
  - 長さ3, 4, 5以上のフレーズを各20,000ペアずつ

Model			Annotated-PPDB
word vectors	n	comp.	
skip-gram	25	+	0.20
PARAGRAM	25	+	0.32*
PARAGRAM	25	RNN	**0.40***†‡
Ganitkevitch et al. (2013)			0.25
word overlap (strict)			0.26
word overlap (lemmatized)			0.20
PPDB+SVR			0.33

まとめ

- PPDBを利用して短いフレーズの言い換え表現を意味構成モデルにより学習
  - PPDBの有効性を確認

- 今後の方向性
  - フレーズベクトルの距離尺度の再考
  - フレーズ構造のほかに係り受け構造の利用
  - より長い文のタスクにおける短い言い換え表現モデルの活用