A Compositional and Interpretable Semantic Space の紹介

須田山

東大 高野研 M1

August 29, 2015

前置き

- ベクトル空間モデル
- 意味合成
- ベクトル表現の解釈可能性

手法

実験: 句の推定

実験: 句の解釈可能性

ベクトル空間モデル

- **コーパスである単語の使われ方をもとに低次元ベクトルとして単語を表現**
- word analogy **タスクで単語の間の** syntactic/semantic **な関係性を捉えられている度合いで評価** [1]

[1] Linguistic Regularities in Sparse and Explicit Word Representations (Levy and Goldberg, 2014)

意味合成

句の類似

- 2 つの句の類似度合いを反映した素性があるとうれしい
  - 言い換え表現の検出
  - 感情分析

意味合成 (*semantic compsotion*)

複数の単語からなる句のベクトル表現を構成語の合成から得る [2]

- **合成演算で行列を用いて形容詞+名詞を詳しく調べた研究** (Baroni and Zamparelli, 2010)
- **文の感情分析** (Socher et al., 2012)[3]

---

[2] Vector-based models of semantic composition(Mitchell and Lapata, 2008)

[3] Semantic Compositionality through Recursive Matrix-Vector Spaces(Socher, Brody, Manning and Ng, 2012)

ベクトル表現の解釈可能性

- **ベクトル表現を用いるシステム: 誤りの分析が難しい**
- **解釈が容易なベクトル表現を得るための工夫**
  - **成分に対する非負制約** (cf. Non-negative Matrix Factorization)
  - **ベクトルが疎である制約**

解釈の容易さをどうやって評価するか?

解釈可能性の評価

Word Intrusion Detection Task[4]

$A \in \mathbb{R}_{+}^{m \times k}$, m: **単語数**, k: **単語ベクトルの次元**

1. **次元** j $(1 \leq j \leq k)$ **の成分の大きさ** $A[:, j]$ **で単語を並べる**: S_0
2. S_0 **から** 6 **つ単語を選ぶ**:
   - **上位** 5 **つ**
   - intruder **を** 1 **つ**:
     - S_0 **の後半に現れる**
     - **別の次元** j' $(1 \leq j' \leq k)$ **の成分の大きい順で上位** 10%**に入る**
3. **選んだ** 6 **単語を無作為に並べて、人間が仲間はずれを選べるかで評価する**

[4] Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding(Murphy, Talukdar and Mitchell, 2012)

前置き

手法

- NNSE におけるベクトル表現の解釈
- CNNSE でのベクトル表現の例
- NNSE/CNNSE の目的関数

実験: 句の推定

実験: 句の解釈可能性

手法

既存のアプローチ

共起行列 疎で次元も大きく、そのままでは扱いにくい

Latent Semantic Analysis(LSA) 成分の解釈に難あり
句をともに構成するような単語の関係を考慮せず

著者らの手法

LSA のような行列分解というアプローチをベースに解釈が容易になるように次の制約を加える:

- 単語のベクトルが疎
- 各成分が非負

Non-negative Sparse Embeddings(NNSE)

- **次元の解釈**
  - **非負制約があるのである次元で大きい値を持つ単語を調べればよい**
- **単語の解釈**
  - **単語ベクトルで大きい値を持つ成分の数は疎である制約のために少なくなる**
  - **少数の大きい成分に対応する次元を調べることで単語の意味が判明する**

CNNSE(Compositional-) **は著者らの既存手法である** NNSE **に意味合成に関する制約を加えたものを指す**

CNNSE によるベクトル表現の例

military	aid	military aid
serviceman	guidance	servicemen
commandos	advice	commandos
military intelligence	assistance	military intelligence
guerrilla	mentoring	guidance
paramilitary	tutoring	advice
anti-terrorist	internships	assistance
conglomerate	award	compliments
giants	awards	congratulations
conglomerates	honors	replies

NNSE

$$\begin{aligned} \underset{A,D}{\arg\min} \quad & \frac{1}{2}\sum_{i=1}^{w} \|X_{i,:} - Ai, : \times D\|_2^2 + \lambda_1 \|A_{i,:}\|_1 \\ \text{subject to} \quad & D_{i,:} D_{i,:}^{\top} \leq 1 \quad i = 1, \ldots, l \\ & A_{i,j} \geq 0 \quad i = 1, \ldots, w \,,\, j = 1, \ldots, l \end{aligned}$$

- **第二項の L_1 正則化項によって疎であるという制約が課される**
- **A, D の一方を固定すると損失関数は凸になる**

CNNSE

句の合成に関する制約を加える:

- **単語** i, j **からなる句** p
- **合成関数** $f\left(u, v\right)$

$$A_{(p,:)} = f\left(A_{(i,:)}, A_{(j,:)}\right)$$

$$\begin{aligned}
\underset{A,D}{\arg\min} \quad & \frac{1}{2}\sum_{i=1}^{w} \|X_{i,:} - Ai, : \times D\|_2^2 + \lambda_1 \|A_{i,:}\|_1 \\
& + \frac{\lambda_c}{2} \sum_{\substack{\text{phrase } p, \\ p=(i,j)}} \left\{A_{(p,:)} - f\left(A_{(i,:)}, A_{(j,:)}\right)\right\}^2 \\
\text{subject to} \quad & D_{i,:} D_{i,:}^{\top} \leq 1 \quad i = 1, \ldots, l \\
& A_{i,j} \geq 0 \quad i = 1, \ldots, w\,,\ j = 1, \ldots, l
\end{aligned}$$

合成関数

f として用いる関数の候補はいくつかある:

(weighted) addition $f(\boldsymbol{u}, \boldsymbol{v}) = \alpha\boldsymbol{u} + \beta\boldsymbol{v}$

multiplication $f(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{v}$

今回は weighted addition **を用いる**:

- **形容詞+名詞の合成に関して有効** (Mitchell and Lapata, 2010)
- **最適化の都合もよい**

CNNSE

合成関数: $f(\boldsymbol{u}, \boldsymbol{v}) = \alpha\boldsymbol{u} + \beta\boldsymbol{v}$

合成の制約を表すのに行列 $B \in \mathbb{R}^{l \times l}$ **を導入**:

- **元の制約**: $B_{(p,i)} = -\alpha, \quad B_{(p,j)} = -\beta$
- **同一の句の合成**: $B_{(p,p)} = 1$
- **名詞+名詞の合成で** A **と同時に** α, β **を最適化するなら** $\alpha = \beta$
- **平均という解釈ができるので著者らは** $\alpha = \beta = 0.5$ **とした**

整理すると損失関数は次のようになる:

$$\arg\min_{A,D} \frac{1}{2}\|X - AD\|_F^2 + \lambda_1\|A\|_1 + \frac{\lambda_c}{2}\|BA\|_F^2$$

$$\|M\|_F = \mathrm{trace}\left(M^\top M\right)$$

最適化

先の損失関数:

$$\arg\min_{A,D} \frac{1}{2}\|X - AD\|_F^2 + \lambda_1\|A\|_1 + \frac{\lambda_c}{2}\|BA\|_F^2$$

上の式を関数の和とみるとそれぞれは凸関数:

- $f(A_0, D) = \frac{1}{2}\|X - A_0 D\|_F^2$
- $g(A_1) = \lambda_1\|A_1\|_1$
- $h(A_2) = \lambda_c\|BA_2\|_F^2$

このタイプの最適化問題は交互乗数方向法 (ADMM) **で解ける**

前置き

手法

実験: 句の推定

- データセットと評価尺度
- 比較対象と結果
- 誤りの分析

実験: 句の解釈可能性

実験: データセット

著者らによる前の研究 (Fyshe et al., 2013) **で得た単語と句に対するベクトル表現を用いた**:

- **コーパス**: ClueWeb09(Callan and Hoy, 2009) **の一部** (16 billion words)
- **単語か形容詞+名詞の形をとる句に次の素性を付加**:
  1. PoS
  2. **依存関係**: Maltparser **から**
- **入力**: **単語-依存関係の共起行列**
- **入力行列の成分**: Positive Pointwise Mutual Information $\mathrm{PPMI}(x;y) = \max\left(0, \log \frac{p(x,y)}{p(x)p(y)}\right)$
- **出力として** SVD **による** 1000 **次元のベクトル**

実験: データセット

上のベクトル表現

- 合わせて 54454 の単語、句に対するベクトル
- 形容詞+名詞からなる句を訓練 $(2/3)$ とテスト $(1/3)$ 集合に分割
- 単語自体は同一のものが双方の集合に現れうる

実験: 句ベクトルの推定

テスト用の句に対するベクトルを cosine **距離で並び替える**: S_0
いずれも上限は 100

median rank accuracy rankaccuracy **を句ごとに計算して** median **をとったもの**

rank accuracy: $100 \times \left(1 - \frac{r}{P}\right)$

- r: S_0 **における正しい句の位置**
- P: **テスト用の句の数**

mean reciprocal rank(MRR) $100 \times \left(\frac{1}{P}\sum_{i=1}^{P}\frac{1}{r}\right)$

- **リストの先頭に近いアイテムをより重視する**

perfect **並び替えの先頭に正しい句があるケースの割合**

実験: 比較対象

w.add(SVD)

- $\hat{X}_{(p,:)} = \alpha X_{(i,:)} + \beta X_{(j,:)}$
- **パラメータ**: α, β

lexfunc(Baroni and Zamparelli, 2010)

- $\hat{X}_{(p,:)} = M_i X_{(j,:)}$
- **パラメータ**: $\{M_i | i \in \mathrm{Ph}\}$

w.add(NNSE)/CNNSE

NNSE **通常の手順で訓練したのち、合成パラメータ α, β を最適化**

CNNSE **著者らの提案手法、合成による損失を同時に考慮**

実験: スコア

Model	Med. Rank	MRR	perfect
w.add(SVD)	99.89	35.26	20%
w.add(NNSE)	99.80	28.17	16%
Lexfunc	99.65	28.96	20%
CNNSE	99.91	40.65	26%

- Median rank accuracy **での比較では大きな差がない**
- MRR **では** 5 point **以上の差**
- w.add(NNSE) **は** CNNSE **と比較しても低く、合成に関する制約を組み込む利点を強調している**

誤りの分析

並び替えで先頭に来た句: 正答と関連性があるかもしれない

そこで正答と各モデルで誤って先頭に現れた句の関連を調べるための補助実験:

1. **すべてのモデルで先頭に誤った句が来るようなものを見つける**
2. Mechanical Turk **のユーザーにそれぞれのモデルで先頭に現れた句を提示する**
3. **もっとも正答と関連があるものを選んでもらう**

- **条件を満たす句のうち無作為に** 200 **件**
- 5 **人のユーザーの過半数で合意が得られたもののみで割合を計算**

誤りの分析

Model	**選ばれた割合**
w.add(SVD)	21.3%
w.add(NNSE)	11.6%
Lexfunc	31.7%
CNNSE	35.4%

- Lexfunc **に比べて** CNNSE **が若干上回っている**

前置き

手法

実験: 句の推定

実験: 句の解釈可能性

- word intrusion task
- **句の表現**
- **句の振る舞い**

interpretability: word intrusion task(再掲)

$A \in \mathbb{R}_{+}^{m \times k}$　**単語数**: m, **単語ベクトルの次元**: k

1. **ある次元** j $(1 \leq j \leq k)$ **の成分** $A[:, j]$ **が大きい順に並べる** (S_0)
2. S_0 **の上位** 5 **つをとる**
3. **さらに** S_0 **から次の条件を満たす単語** (*intruder*) **を** 1 **つ選ぶ**:
   - 3.1 S_0 **の後半に現れる**
   - 3.2 **別の次元** j' $(1 \leq j' \leq k)$ **の成分の大きい順で上位** 10% **に入る**
4. **選んだ** 6 **単語を無作為に並べて、人間が仲間はずれを選べるかで評価する**

interpretability: word intrusion task による比較

Method	Intruders detected	Mturk Agreement
SVD	17.6%	74%
NNSE	86.2%	94%
CNNSE	88.9%	90%

- intruder の検出率では CNNSE がもっともよい (“Intruders detected”)
- 評価者の間では NNSE のほうが intruder について合意がとれている (“Mturk Agreement”)
- SVD は偶然による検出率に近い $(1/6)$

interpretability: 句の表現の coherence

1. 句の表現でスコアが大きい次元を 10 個とってくる
2. それらの次元で大きなスコアを持つ単語のリスト (*interpretable summarization*) をとってくる
3. ユーザーは各モデルで得られた、句の interpretable summarization が複数与えられ、次のいずれかを選ぶ:
   - もっとも関連のある単語のリストとしていずれかを選ぶ
   - いずれも関連はない
   - どの単語のリストも同程度に句と関連がある

interpretability: **句の表現の** coherence

Model	**割合**
CNNSE	54.5%
NNSE	29.5%
Both	4.5%
Neither	11.5%

句の振る舞いに関する比較

adjective-noun phrase similarity dataset

- 108 の句の対からなる
- 18 人の被験者によってつけられたスコアの平均を対の類似度

- 類似度を元に 3 分割
- $12/108$ をパラメータの調整に
- 句の cosine similarity と上のスコアとの間の相関を調べる: Spearman's rho

spearman's rho

二つの変数 X_i, Y_i の順位を示す変数 x_i, y_i に対して相関を調べる
($x_1, \ldots, x_n\, y_1, \ldots, y_n$ はそれぞれ $1, \ldots, n$ の permutation)

$$\rho \in [-1, 1] = 1 - \frac{6 \sum_i (x_i - y_i)^2}{n\,(n^2 - 1)}$$

句の振る舞いに関する比較

Model	rho
w.add(SVD)	0.5377
w.add(NNSE)	0.4469
Lexfunc	0.1347
CNNSE	0.5923

- Lexfunc **の相関が低いのは** lexfunc **の元研究** (50) **と比べて訓練に用いる句が少なかった** (39) **ためかもしれない**
- NNSE **と** CNNSE **の比較から目的関数で合成を考慮することを利点として挙げている**

句の振る舞いに関する比較: 例

例: large number/great majority

- similarity score: 5.61(2 **つの句は類似**)

large number		great majority	
0.0831	assortment bevy diverse range	0.1271	candidacy candidate caucus
0.0601	First address complete	0.1169	entire entire entire
⋮	⋮	⋮	⋮
0.0156	average	0.0417	dud good great
0.0132	arithmetic binomial boolean	0.0386	Best Democratic championship

句の振る舞いに関する比較: 分析

上の例は人間によって類似している (score: 5.61) **とされたにもかかわらず共通する次元が現れていない**

- "majority" **政治の文脈での用いられ方から影響を受けている**
  "candidate caucus, . . ."
- "large number" **順番の記述の影響を受けている**
  "First addresss, complete address . . ."

句の振る舞いに関する比較: まとめ

- 例を調べることで著者らのモデルでは多義語の扱いが難しいことがわかる
- 解釈が容易なベクトル表現によってこのような分析が可能に