

# Joint Word Representation Learning Using a Corpus and a Semantic Lexicon (AAAI'16)

Danushka Bollegala (Liverpool U.)

Takanori Maehara (Shizuoka U.)

Ken-ichi Kawarabayashi (NII)

2016年8月10日, ERATO 感謝祭 III @ NII

# 自然言語処理の機械学習的タスク

(1) 自然言語処理特有タスク：形態素解析，構文解析

(2) 機械学習一般タスク，データが自然言語：

- 次にくる単語を予想したい（prediction）
- 単語が肯定的か否定的か知りたい（classification）
- 似たような意味の単語を集めたい（clustering）

# 単語のベクトル表現

機械学習アルゴリズムの入力： $n$ 次元ベクトル

⇒ 単語をベクトルで表現すれば一般的手法が使える

Q. どうやってベクトルを構成する？

A. いろいろある：

- ① 単語数次元空間の単位ベクトル

$$i \mapsto \mathbf{e}_i \in \mathbb{R}^W$$

- ② 周辺共起を数える（分布仮説）

$$i \mapsto \sum_{j:i \text{ cooccurred}} X_{ij} \mathbf{e}_j \in \mathbb{R}^W$$

- ③ 周辺共起から予測問題を作る（ $2 +$ 次元圧縮）

$$i \mapsto \mathbf{w}_i \in \mathbb{R}^d$$

分布仮説「単語の意味は周辺共起で決まる」

# 予測表現例 GloVe [Pennington'14]

$X_{ij}$  : 単語  $i$  と  $j$  が共起した回数 (データから観測)  
各単語  $i$  にベクトル  $w_i$  とスカラー  $b_i$  を割り当てて

$$(w_i, w_j) + b_i + b_j \sim \log X_{ij}$$

を満たすように  $\{w_i, b_i\}_{i \in \text{words}}$  を探す  
(適当に重みづけして最小二乗問題を解く)

実際は共起に向きを入れて変数を倍にする

see: <http://www-nlp.stanford.edu/pubs/glove.pdf>

# 分散表現の面白さ

Skip-Gram, CBoW, GloVe など最近の単語表現の特徴

「単語の差」が「意味」を表現しているっぽい

$$w(\text{king}) - w(\text{man}) \sim w(\text{queen}) - w(\text{woman})$$

- 新しいアプリケーション
  - 観光：凱旋門 - パリ + 東京
  - 料理：麻婆豆腐 - 豆腐 + 茄子
- 証明はまだない

「より良い表現」の需要が高まっている

# より良い表現を作るには？

- 予測モデルを改善する  
(skip-gram, CBOW, GloVe, ...)  
... **red ocean**; 結構つらい・報われない
- データ（コーパス）を増やす  
(wikipedia dump, gigaword, clawled data, ...)  
「ウェブからとれる英文」 以外はつらい
- 高品質データと組み合わせる ← **new direction**

# 高品質データ：辞書 (eg: WordNet)

- データ量は少ない・更新頻度も少ない
- 専門家が整備しているので高品質

## WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

### Noun

- [S: \(n\) cat](#), [true cat](#) (feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats)
- [S: \(n\) guy](#), [cat](#), [hombre](#), [bozo](#), [sod](#) (an informal term for a youth or man) "*a nice guy*"; "*the guy's only doing it for some doll*"; "*the poor sod couldn't even buy a drink*"
- [S: \(n\) cat](#) (a spiteful woman gossip) "*what a cat she is!*"
- [S: \(n\) kat](#), [khat](#), [qat](#), [quat](#), [cat](#), [Arabian tea](#), [African tea](#) (the leaves of the shrub *Catha edulis* which are chewed like tobacco or used to make tea; has the effect of a euphoric stimulant) "*in Yemen kat is used daily by 85% of adults*"

# 本研究のアイデアと結果

コーパス + 辞書 = 高品質表現

- 基本的にコーパスのみより高性能  
（使う辞書のデータに微妙に依存）
- 既存のコーパス + 辞書手法よりも高性能
- レア単語やコーパスが小さい場合に特に有効

感想：当たり前の結果がちゃんと出た



# これまでの我々の研究との関連

「比較的小さいコーパスでも高性能」を求めてきた

- Learning word representations from relational graphs (AAAI'15)  
 どういう文脈で共起したかも使う
- Embedding semantic relations into word representations (IJCAI'15)  
 ベクトル同士の差が意味をもつよう制約する
- Unsupervised cross-domain word representation learning (ACL'15)  
 ドメインごとの意味の違いを吸収する

# 提案手法

# 提案手法のアウトライン

目標：コーパスと辞書を組み合わせる

コーパスからの学習  $\Rightarrow$  GloVe 目的関数  $J_C$

辞書からの学習  $\Rightarrow$  正則化項  $J_S$

これらを足して最適化する： $J = J_C + \lambda J_S$

# 目的関数

GloVe 目的関数  $J_C$

$$J_C = \sum_{ij:\text{cooccurred}} f(X_{ij})((w_i, w_j) + b_i + b_j - \log X_{ij})^2$$

辞書正則化項  $J_S$

$$J_S = \sum_{ij:\text{related}} \|w_i - w_j\|^2$$

変数を固定すると凸 (jointly convex) なので交互最適化  
それぞれの最適化は確率勾配法 (AdaGrad)

# 実験

# 扱うタスク

- similarity prediction  
(book,magazine)  $\leq$  (book,text)  
⇒ 求めた表現のコサイン類似度を比較
- propotional analogy  
king : man = ?????? : woman  
⇒  $a - b + d$  とコサイン類似度最大

これらについて，標準的なデータセットで評価

- RG, MC, RW, SCWS, MEN ... similarity prediction
- sem, syn, total, SemEval ... propotional analogy

# 辞書の取り入れ方

辞書上の関係にはいろいろなパターンがある

- synonym (同義語) big – large
- antonym (反義語) old – young
- hypernym (上位語) building – house
- hyponym (下位語) house – building
- holonyms (全体) animal – cat
- meronyms (部分) cat – animal

とりあえず色々使って、どれが結果に効く  
(どの関係が性能に効くのか、という興味もある)

# 実験結果（コーパスのみと比較）

Table 1: Performance of the proposed method with different semantic relation types.

Method	RG	MC	RW	SCWS	MEN	sem	syn	total	SemEval
corpus only	0.7523	0.6398	0.2708	0.460	0.6933	61.49	66.00	63.95	37.98
Synonyms	<b>0.7866</b>	<b>0.7019</b>	0.2731	<b>0.4705</b>	<b>0.7090</b>	61.46	<b>69.33</b>	<b>65.76</b>	<b>38.65</b>
Antonyms	0.7694	0.6417	0.2730	0.4644	0.6973	61.64	66.66	64.38	38.01
Hypernyms	0.7759	0.6713	0.2638	0.4554	0.6987	61.22	68.89	65.41	38.21
Hyponyms	0.7660	0.6324	0.2655	0.4570	0.6972	61.38	68.28	65.15	38.30
Member-holonyms	0.7681	0.6321	0.2743	0.4604	0.6952	<b>61.69</b>	66.36	64.24	37.95
Member-meronyms	0.7701	0.6223	0.2739	0.4611	0.6963	61.61	66.31	64.17	37.98
Part-holonyms	0.7852	0.6841	0.2732	0.4650	0.7007	61.44	67.34	64.66	38.07
Part-meronyms	0.7786	0.6691	<b>0.2761</b>	0.4679	0.7005	61.66	67.11	64.63	38.29

- 全般的に Synonyms (同義語) を使うとよく, holonyms (全体), meronyms (部分) が続く
- レア単語の similarity prediction (**RW**) において holonyms/meronyms が特に有効  
⇒ コーパスでカバーできない部分を辞書でカバー



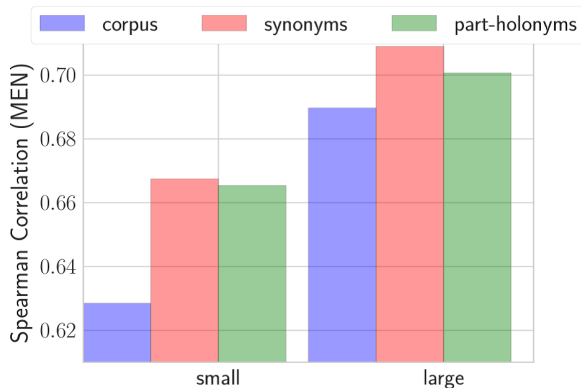
# 実験結果（既存手法と比較）

Table 2: Comparison against prior work.

Method	RG	MEN	sem	syn
RCM	0.471	0.501	-	29.9
R-NET	-	-	32.64	43.46
C-NET	-	-	37.07	40.06
RC-NET	-	-	34.36	44.42
Retro (CBOW)	0.577	0.605	36.65	52.5
Retro (SG)	0.745	0.657	45.29	65.65
Retro (corpus only)	0.786	0.673	61.11	68.14
Proposed (synonyms)	<b>0.787</b>	<b>0.709</b>	<b>61.46</b>	<b>69.33</b>

- 最善の結果
- レトロフィッティング [Faruqui'15] よりも同時最適化（提案法）のほうが性能が上

# 実験結果（コーパスサイズを限定）



- small = 10%をランダムサンプルしたコーパス
- コーパスサイズが小さい場合に特に有効

# まとめ

コーパス + 辞書 = 高品質表現

- コーパス項を辞書項で正規化して最適化
- 交互最適化・確率勾配法
  
- コーパスのみを基本的に上回る
- 既存のコーパス + 辞書手法を統合的に上回る
- コーパスが小さい場合に特に有効