複数の特徴ベクトルのクラスタリングに基づく単語の意味の弁別

九岡　佑介　　白井　清昭　　中村　誠
北陸先端科学技術大学院大学 情報科学研究科
{s0610032, kshirai, mnakamur}@jaist.ac.jp

1 はじめに

本論文はコーパスに出現する単語の意味を自動的に弁別する手法について述べる．通常の語義曖昧性解消では，単語の意味 (語義) を辞書などによってあらかじめ定義し，特定の文脈中に出現した単語の意味を定義された語義の中から選択する．ところが，単語の意味は日々変化し，新しい意味や用法も生まれている．あらかじめ語義を定義するというアプローチではこのような単語の意味の変化に対応することができない．本研究では，コーパス中に出現する単語を特徴ベクトルで表現し，教師なしクラスタリングによって同じ意味を持つ単語をひとつのクラスタにまとめることで，既存の辞書に依らずに単語の意味を弁別することを目的とする．単語の意味の自動弁別は，単語の新しい意味や用法の自動的な発見や辞書編纂作業のサポートに応用できる．

本研究と同様に，クラスタリングによって単語の意味を弁別する試みがいくつか行われている．Schütze は，文脈内に出現する単語から構成されるベクトルをクラスタリングすることで単語の意味を弁別する手法を提案している [8]．英語の動詞を対象とした同様の試みが Fukumoto らによって報告されている [3]．Bordag は，対象単語を含みかつ互いによく共起する単語の 3 つ組をコーパスから生成し，それらをクラスタリングすることで単語の意味を推定する手法を提案している [2]．Pantel らは，複数の種類の単語をクラスタリングすることで同値関係にある単語の集合を自動的に獲得する手法を提案している [7]．本研究も，これらの先行研究と同様に，単語を特徴ベクトルで表現し，その特徴ベクトルをクラスタリングすることで単語の意味を弁別するが，単語を複数のタイプの特徴ベクトルで表現し，それらを併用してクラスタリングを行う点に特徴がある [5]．

2 提案手法

提案手法における処理の流れは以下の通りである．まず，意味を弁別する対象単語を w とする．また，コーパスにおける w の個々の出現をインスタンスと呼び，w_i で表わす．w_i を特徴ベクトル $\vec{f_i}$ で表現し，$\vec{f_i}$ を教師なしクラスタリングによっていくつかのクラスタに分類する．同じ意味を持つインスタンス w_i がひとつのクラスタにまとめられたとみなすことで単語の意味を弁別する．以下，2.1 項では単語の特徴ベクトルについて述べ，2.2 項でクラスタリング手法について述べる．

2.1 特徴ベクトル

本研究では単語の出現を 4 種類の特徴ベクトルを用いて表現する．

2.1.1 隣接ベクトル

隣接ベクトル $\vec{n_i}$ は，w_i の直前または直後に現われる単語で w_i を特徴付けるベクトルである．具体的には，w_i の前後 1 語の単語の出現形ならびに品詞をベクトルの要素とする．品詞は茶筌[1]の品詞体系を用いる．ベクトルの各要素の重みは，該当する要素が w_i の前後に現われる場合は 1，それ以外は 0 とする．

2.1.2 文脈ベクトル

文脈ベクトル $\vec{c_i}$ は，w_i の周辺に現われる単語で w_i を特徴付けるベクトルである．すなわち，ある 2 つのインスタンス w_i と w_j に対し，それらの周辺に同じ単語が出現していれば，w_i と w_j が似ている (同じ意味を持つ) とみなす．ただし，一般に文脈ベクトルはスパースになることが予想される．そのため，w_i と w_j の周辺に出現する単語に重なりがなく，クラスタリングの手がかりとなる情報が得られない可能性もある．そこで，周辺文脈に出現する自立語だけでなく，その関連語を $\vec{c_i}$ の要素に追加することにより，文脈ベクトルの過疎性を補完する．

まず，コーパスから，単語 c_k を行，文書 d_l を列とする共起行列 $\mathbf{A_c}$ を作成する．$\mathbf{A_c}$ の要素 $a_{i,j}$ は，単語 c_k が文書 d_l に出現した回数とする．コーパスとして「Yahoo! 知恵袋」コーパスを用いた．このコーパスは Yahoo!知恵袋 [2]に掲載された 45,725 組の質問と回答から構成される約 500 万語のテキストである．ここでは，質問と回答の組を 1 つの文書とみなした．共起行列 $\mathbf{A_c}$ の行を構成する単語 c_k として，コーパスにおける出現頻度の上位 20,000 個の自立語を選定した．ただし，コーパス全体の 1 割以上の文書に出現する単語は一般的すぎるとみなして除い

[1] http://chasen.naist.jp/hiki/ChaSen/
[2] http://chiebukuro.yahoo.co.jp/

た．次に，行列 $\mathbf{A_c}$ に対して LDA(Latent Dirichlet Allocation) [1] を適用した．LDA は，PLSI(Probabilistic Latent Semantic Indexing) [4] と同様に，トピックと単語の関連性を表わす確率パラメタを学習する手法である．ここでは，LDA によって得られる確率パラメタ $P(c_k|z_m)$ を利用する．z_m は隠れ変数であり，直観的にはトピックを表わす．したがって，$P(c_k|z_m)$ はトピック z_m と単語 c_k の関連の強さを表わす．LDA では隠れ変数の数はあらかじめ定める必要があり，本論文では 50 とした．次に，各トピック z_m に対し，そのトピックと最も関連性の高い 300 個の単語の集合 Z_m を作成する．具体的には，式 (1) の値が高い上位 300 個の単語をトピック毎に選定し，Z_m の要素とする．

$$\log \frac{P(c_k|z_m)}{P(c_k)} \tag{1}$$

以上の前処理を経て，インスタンス w_i の文脈ベクトル $\vec{c}_i$ を以下のように作成する．文脈ベクトルは単語を要素とするベクトルとする．もし，w_i の周辺に自立語 c_{ij} が出現したなら，$\vec{c}_i$ 中の c_{ij} の重みを 1 とする．さらに，c_{ij} が Z_m に含まれているなら，Z_m 中の残りの単語について，$\vec{c}_i$ 中のその単語の重みを 0.5 にする．それ以外の単語の重みは 0 とする．$\vec{c}_i$ では，w_i の周辺に出現しない単語でも，周辺に出現する単語と同じトピックを持つとみなせる単語に対しては正の重みを与えることで，特徴ベクトルの過疎性を補完している．

2.1.3 連想ベクトル

連想ベクトル $\vec{a}_i$ は，文脈ベクトルと同じく，w_i の周辺に現われる単語で w_i を特徴付けるベクトルである．ただし，ベクトルの過疎性を補完するために，事前にコーパスから作成された単語の共起行列を用いる点が異なる．

まず，「Yahoo!知恵袋」コーパスから単語の共起行列 $\mathbf{A_a}$ を作成する．行と列はともに単語で，コーパスにおける出現頻度上位 10,000 の単語とする．ただし，コーパス全体の 1 割以上の文書に出現する単語は一般的すぎるとみなして除く．行列 $\mathbf{A_a}$ の要素 $a_{i,j}$ は，単語 c_i と単語 c_j が同じ文書に共起した回数とする．次に，$\mathbf{A_a}$ の各列 $(a_{1,j}, \cdots, a_{10000,j})^T$ を単語 c_j の共起ベクトル $\vec{o}(c_j)$ とする．すなわち，頻度の上位 10,000 語の単語に対して 10,000 次元からなる共起ベクトル $\vec{o}(c_j)$ を得る．

以上の前処理を経て，連想ベクトル $\vec{a}_i$ を式 (2) のように定義する．

$$\vec{a}_i = \sum_{c_j \in context} \vec{o}(c_j) \tag{2}$$

すなわち，$\vec{a}_i$ は，w_i の周辺に現われる自立語 c_j に対する共起ベクトル $\vec{o}(c_j)$ の和と定義する．ただし，w_i の周辺には現われるが，共起ベクトル $\vec{o}(c_j)$ が得られていない (出現頻度の上位 10,000 の単語集合に含まれない) 単語は無視される．また，クラスタリングの際には，式 (2) のベクトルを大きさが 1 になるように正規化する．

2.1.4 トピックベクトル

トピックベクトル $\vec{t}_i$ は，PLSI によって推定されるトピックによって w_i を特徴付けるベクトルである．まず，文脈ベクトルと同様に，単語を行，文書を列とする共起行列 $\mathbf{A_c}$ を作成する．ここでは $\mathbf{A_c}$ に対して PLSI を適用し，トピックと単語の関連性を表わす確率パラメタを学習する．PLSI の隠れ変数の数 M は 50 と設定した．

次に，インスタンス w_i に対し，w_i を含む文書を d_i とおく．d_i を PLSI の学習データに含まれない未知の文書とみなして，確率パラメタ $P(z_m|d_i)$ を Folding-in [4] と呼ばれる EM アルゴリズムによって推定する．最後に，w_i のトピックベクトル $\vec{t}_i$ を式 (3) のように定義する．

$$\vec{t}_i \ = \ (\ P(z_1|d_i), \cdots, P(z_M|d_i)\ )^T \tag{3}$$

すなわち，トピックベクトル $\vec{t}_i$ は，PLSI の隠れ変数 z_j を要素とし，$P(z_j|d_i)$ を重みとする特徴ベクトルである．$\vec{t}_i$ は，正確には w_i そのものよりも w_i を含む文書 d_i の特徴を表わすベクトルである．したがって，$\vec{t}_i$ によるクラスタリングは，同じトピックの文書に現われる単語は同じ意味を持つという観点で単語の意味を弁別しているとみなせる．

2.2 クラスタリング

本研究では教師なしクラスタリングアルゴリズムである k-means 法 [6] を用いて w_i のクラスタリングを行う．k-means 法ではあらかじめクラスタの数 k を定義する必要がある．クラスタの数は，ここでは単語の語義の数に相当する．本来なら単語が持つ語義の数も自動的に決定するべきであるが，本論文では仮に $k = 10$ とした．語義の数をどのように決定するかは今後の重要な課題である．また，特徴ベクトル間の類似度はコサイン類似度で測った．

2.2.1 複数の特徴ベクトルの併用

本研究では 2.1 項で述べた 4 種類の特徴ベクトルをクラスタリングすることで単語の意味を弁別する．ところが，予備実験の結果，意味を最もよく弁別できる特徴ベクトルは単語によって異なることがわかった (詳細は 3.2 項で述べる)．このことは，意味を正確に弁別できる素

性は対象単語によって異なることを示唆する．ここでは，複数の特徴ベクトルによるクラスタリングの結果を比較し，最も良い結果が得られると思われる特徴ベクトルを単語毎に選択することで，クラスタリングの精度を向上させる手法について述べる．

クラスタリングの結果を $C = \{C_i\}$ とおく．C_i はひとつのクラスタを表わす．C_i を構成する単語を w_{ij}，その特徴ベクトルを $\vec{v}_{ij}$ とする．クラスタリングの良さを表わす評価尺度 $E(C)$ を式 (4) のように定義する．

$$E(C) = \sum_i coh(C_i) \quad (4)$$

$$coh(C_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} sim(\vec{v}_{ij}, \vec{g}_i) \quad (5)$$

式 (5) で定義した $coh(C_i)$ (cohesiveness) は，クラスタ C_i 内の要素がどれだけ互いに類似しているかを定量的に測る尺度である．すなわち，$coh(C_i)$ は，クラスタ内の要素 w_{ij} の特徴ベクトル $\vec{v}_{ij}$ と C_i の重心 $\vec{g}_i$ の類似度 $(sim(\vec{v}_{ij}, \vec{g}_i))$ の平均である．N_i は C_i の要素数を表わす．また，$E(C)$ は各クラスタの $coh(C_i)$ の重み付き平均と定義する (式 (4))．本研究では，$E(C)$ の大きさが大きければ大きいほど，すなわちクラスタ内の各要素がそのクラスタの重心と近い位置にあればあるほど，クラスタリングが成功しているとみなす．そこで，隣接ベクトル $\vec{n}_i$, 文脈ベクトル $\vec{c}_i$，連想ベクトル $\vec{a}_i$，トピックベクトル $\vec{t}_i$ を用いてクラスタリングを行い，それぞれのクラスタリングの結果を $E(C)$ で評価し，最も大きい値を持つ特徴ベクトルのクラスタリングの結果を最終的なクラスタとする．

ところが，予備実験の結果，$E(C)$ はクラスタリングの良さを表わす指標として不適切であることがわかった．なぜなら，ベクトル間の類似度の値が特徴ベクトル毎に大きく異なるためである．例えば，3 節の実験において，2 つの隣接ベクトルの類似度は 0.006 程度であるのに対し，文脈ベクトルの場合は 0.0003 程度であった．このため，$E(C)$ によって異なる特徴ベクトルによるクラスタリングの結果を比較しようとしても，ベクトル間の類似度の平均値が大きい特徴ベクトルが常に選択される．この問題を解決するために，$coh(C_i)$ を式 (6) のように修正した．

$$coh(C_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} rel\text{-}sim(\vec{v}_{ij}, \vec{g}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} \frac{sim(\vec{v}_{ij}, \vec{g}_i)}{\max_j sim(\vec{v}_{ij}, \vec{g}_i)} \quad (6)$$

式 (6) における $rel\text{-}sim$ は $\vec{v}_{ij}$ と $\vec{g}_i$ の相対的な類似度を表わす．すなわち，$rel\text{-}sim(\vec{v}_{ij}, \vec{g}_i)$ の分子は $\vec{v}_{ij}$ と $\vec{g}_i$ の類似度，分母はクラスタ内で最も重心との類似度が大きい特徴ベクトルと重心との類似度である．クラスタ内の要素と重心の近さを測る際，重心との近さを相対的に評価することで，特徴ベクトルの類似度の平均値にばらつきがあってもクラスタリングの良さをある程度正しく比較できる．

3 評価実験

3.1 実験手順

まず，単語の意味を弁別する対象単語として以下の 23 個の単語を選定した．

> モデル，ネタ，カバー，ウイルス，ソース，肉，サービス，地方，アルバム，コード，自分，場合，時間，意味，電話，一緒，目，以前，代，顔，系，郵便，反応

次に，各対象単語に対し，「Yahoo!知恵袋」コーパスの中から 100 インスタンスをランダムに選択した．これらを提案する 4 つの特徴ベクトルで表現し，k-means 法によってクラスタリングした．

正解データとして，コーパス中のインスタンスに人手で語義を付与した．語義は岩波国語辞典の中分類を基準に定義した．これは比較的荒い意味分類である．また，岩波国語辞典に該当する語義がない場合は新しい語義を定義した．語義の数は対象単語毎に異なるが，2〜6 個の範囲にある．

クラスタリングの評価尺度として Purity と Inverse Purity(I-Purity) を用いた．それぞれの定義は以下の通りである．

$$Purity(C) = \sum_i \frac{|C_i|}{N} \cdot C_i\text{の最大適合率} \quad (7)$$

$$I\text{-}Purity(C) = \sum_j \frac{|C'_j|}{N} \cdot C'_j\text{の最大再現率} \quad (8)$$

式 (7) における「最大適合率」とは，クラスタ C_i に含まれるインスタンスの語義のうち最も数が多いものの割合である．また，N はクラスタリングの対象となる単語の総数である．すなわち，Purity は，作成されたクラスタがどれだけ同じ語義を持つインスタンスを含むかを評価している．一方，式 (8) における C'_j は真のクラスタ (同じ語義ラベルが与えられたインスタンスの集合) であり，「最大再現率」は C'_j に含まれるインスタンスを自動作成されたクラスタで分類したとき，最も数が多いクラスタ

に属するインスタンスの割合である．すなわち，Inverse Purity は，真のクラスタに属するインスタンスがどれだけ同じクラスタに分類されたかを評価している．

k-means 法は初期のクラスタをランダムに与える．そのため，実験ではクラスタリングを 10 回行い．Purity と Inverse Purity の平均を求めた．

3.2 結果

まず，インスタンス w_i を単独の特徴ベクトルで表現したときのクラスタリングの結果を表 1 に示す．

表 1: 特徴ベクトルを単独で用いたときの結果

	隣接	文脈	連想	トピック
Purity	0.774	0.736	0.779	0.768
I-Purity	0.270	0.217	0.297	0.282

表 1 の値は，23 個の対象語の平均の Purity と Inverse Purity である．4 つの特徴ベクトルのうち，連想ベクトルでクラスタリングしたときが Purity，Inverse Purity ともに最も高かった．

次に，23 個の対象単語について，最も高い Purity または Inverse Purity が得られた特徴ベクトルを調べた．その結果を表 2 に示す[3]．

表 2: 最も良い結果が得られた特徴ベクトルの内訳

	隣接	文脈	連想	トピック
Purity	11	0	7	6
I-Purity	7	0	11	5

文脈ベクトル以外の 3 つの特徴ベクトルについては，最も良いクラスタリング結果が得られる単語の数が均等に分かれている．この結果，クラスタリングに適した特徴ベクトルは単語毎に異なることがわかった．

2.2.1 で述べた手法，すなわち 4 つの特徴ベクトルのクラスタリング結果から式 (4),(6) で得られる評価値 $E(C)$ が最も高い結果を選択したとき，Purity は 0.791，Inverse Purity は 0.312 となった．これらはいずれも表 1 の単独の特徴ベクトルの評価値を上回る．この結果，提案手法による複数の特徴ベクトルの併用が有効であることがわかった．ただし，実際に選択されたのは隣接ベクトルとトピックベクトルのいずれかであり，単独で最も正解率の高い連想ベクトルが選ばれることはなかった．クラスタリング結果の評価指標 $E(C)$ の更なる改良が必要である．

[3]Purity の欄の合計が 23 より多いのは，Purity が最大となる特徴ベクトルが 2 つあるときを重複して数えているためである．

4 おわりに

本研究では，単語のインスタンスをいくつかの特徴ベクトルで表現し，それらをクラスタリングすることで，既存の辞書に依らずに単語の意味を弁別する方法を提案した．最後に今後の課題について述べる．まず，現在はクラスタの数 (語義の数) を事前に与えているが，これを自動的に決定する方法について検討する必要がある．また，提案手法は，得られたクラスタの解釈，すなわちクラスタが単語のどのような意味を表わすかについては明らかにしていない．この問題に対し，クラスタと既存の辞書の語義とを自動的に対応付けたり，あるいはクラスタが既存の辞書のどの意味にも該当しない新語義を表わすかを自動的に判別することで，クラスタが表わす単語の意味の解釈を行うことを検討している．

参考文献

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.

[2] Stefan Bordag. Word sense induction: Triplet-based clustering and automatic evaluation. In *Proceedings of the EACL*, pp. 137–144, 2006.

[3] Fumiyo Fukumoto and Jun'ichi Tsujii. Automatic recognition of verbal polysemy. In *Proceedings of the COLING*, pp. 762–768, 1994.

[4] Thomas Hofmann. Probabilistic latent semantinc indexing. In *Proceedings of the SIGIR*, pp. 50–57, 1999.

[5] 九岡佑介. コーパスからの単語の意味の発見. Master's thesis, 北陸先端科学技術大学院大学, 3 2008.

[6] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics and Probability*, pp. 281–297, 1967.

[7] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 613–619, 2002.

[8] Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, Vol. 24, No. 1, pp. 97–123, 1998.