Factorization Machines を用いた未知の固有表現分類

平田 亜衣　　小町 守
首都大学東京 システムデザイン研究科
{hirata-ai@ed, komachi@}tmu.ac.jp

1 はじめに

固有表現認識（Named Entity Recognition）とは，情報抽出の技術の一つであり，テキストから人名，地名，組織名などの固有表現と呼ばれる表現を自動で認識する処理である．固有表現認識は自然言語処理の他のタスクでも重要な処理であり，文書要約や質問応答などの処理でも必要不可欠である．

これまで，教師あり学習における固有表現認識の研究が行われてきたが，学習コーパスとして大量のタグ付きデータが必要である．しかし，固有表現認識の課題として，日々新しい固有表現が登場するので，学習コーパスに出現しなかった固有表現に対しても正しい予測をする必要がある．

固有表現認識では素性テンプレートを用いた系列ラベリングによるアプローチ [1] が主流である．素性テンプレートを用いれば，固有表現の前後の単語を素性の組み合わせで表すことができるが，素性空間が膨大でとても疎になってしまい，未知の固有表現に対して頑健性が低いという問題がある．そこで，この問題を解決するために，本研究では Factorization Machine を用いて素性の間の関係性を考慮して固有表現認識を行うことを提案する．本研究では学習コーパスに出現しない固有表現に対して正しい予測をすることで実験，比較を行い，Factorization Machines を用いることでスパースな素性と少ない次元数で先行研究と同程度の精度を達成した．

2 関連研究

一般的な日本語固有表現認識の設定では Conditional Random Fields（CRF）[2] や Support Vector Machines（SVM）[3] を用いた手法がよく使用されている．これらの研究では素性テンプレートによる組み合わせ素性の展開と学習が行われているが，組み合わせ素性同士は独立して扱うため，学習データで出現しなかった固有表現に対して頑健ではない．

少ないシードから未知の固有表現を当てるという観点での研究では，Collins ら [4] がタグのついていないデータから co-training を行うことで学習を行っている．これらの半教師あり手法は，組み合わせ素性を陽に展開することで扱うことが可能であるが，適切な正則化ができないという問題点がある．

Primadhanty ら [5] は学習コーパスに出現しなかった未知の固有表現を認識するタスクに取り組んだ．この研究では対数線形モデルを改良しており，前後の文脈，文字種情報などの素性からスコア関数を定義し，テンソルとして構成された重みを行列化し特異値分解（SVD）を用いて核ノルムを計算し，正則化項として用いることでスパースな素性の組み合わせも考慮し，L1，L2 正則化よりも高い精度が得られることを報告している．この手法は組み合わせ素性を扱うために特化した正則化を行っているが，特異値分解が最もよい正則化であるとは限らない．提案手法も行列分解を行うことでスパースな素性を扱うが，分類精度を直接向上させるように分解し，かつ尤度最大化ではなくマージン最大化によって学習を行う点が異なる．

3 Factorization Machines

Factorization Machines [6] とは Support Vector Machine (SVM) と行列分解 (Matrix Factorization) を組み合わせたモデルであり，スパースなデータに対応することができるモデルである．

相互作用の次元を $d = 2$ とした場合の Factorization Machines の予測式は

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j \quad (1)$$

で表される．式 1 の n は素性の次元数，x_i は素性 x の i 番目の次元を表している．線形モデルと同様に，$w \in \mathbb{R}^n$ で表される w は式 1 での重みベクトルであり，第 1 項目の $w_0 \in \mathbb{R}$ はバイアス項，第 2 項目の w_i

は x_i の重みを表している．そして第3項目は v_i, v_j の変数の交互作用をモデルに組み込んでいる．$\langle \cdot, \cdot \rangle$ はサイズ k の2つのベクトルの内積であり，

$$\langle v_i, v_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{2}$$

で表される．v_i は $\mathbf{V} \in \mathbb{R}^{n \times k}$ で表される行列の i 番目の要素であり，k は行列分解したあとの次元数を表すハイパーパラメータである．

今回の実験では2値分類を行うが，Factorization Machines で2値分類を行うときはヒンジロスを計算し，最適化を行う．最適化には Markov Chain Monte Carlo (MCMC) や Stochastic Gradient Descent (SGD) などで推定を行う．

行列分解された $\mathbf{V}$ を用いて交互作用の重みを内積で計算することで，$n \times k$ のサイズの計算をするだけでよく，考慮する交互作用の数が増えても計算量が増えない利点がある．

また，多項式カーネルの SVM では素性間の相互関係が独立であるが，Factorization Machines ではテンソル分解を用いる手法と同様，行列分解された低次元の行列を用いることで素性間の相互関係を学習できるという違いがある．Factorization Machines では学習コーパスに出現しない素性の組み合わせも考慮できるので，今回のタスクの学習コーパスに出現しなかった未知の固有表現をうまく認識できると期待される．

4 未知の固有表現分類実験

今回のタスクとして，学習コーパスに出現しない固有表現の認識を行う．比較手法として，対数線形モデル，Primadhanty らの素性をテンソルとして用いた対数双線形モデル（核ノルム正則化）と多項式カーネルを用いた SVM，Factorization Machine を比較する．

4.1 データ

今回使用するデータは Primadhanty らが作成したデータを使う．

Primadhanty らのデータは CoNLL-2003 をもとにして，テストデータと開発データから学習データに出現した固有表現の候補を取り除いたものである．このデータには，人名（PER），地名（LOC），組織名（ORG），その他の固有表現（MISC），固有表現以外（O）のタグが固有表現の候補となるフレーズに割り当てられている．学習データとテストデータ，開発データのタグの割合を表1に示す．

表 1: Primadhanty らのデータのタグごとの割合（括弧内はユニークな固有表現の数）

	学習データ	開発データ	テストデータ
PER	6,516 (3,489)	1040 (762)	1,342 (925)
LOC	6,159 (987)	176 (128)	246 (160)
ORG	5,721 (2,149)	400 (273)	638 (358)
MISC	3,205 (760)	177 (142)	213 (152)
O	36,673 (5,821)	951 (671)	995 (675)

4.2 素性

今回行う実験の素性は Primadhanty らの実験と同様の素性を用いる．大きく分けて文脈素性と12種類の文字種素性があり，表2の通りである．

4.3 ツール

対数線形モデル，多項式カーネルを用いた SVM は scikit-learn（Version 0.17）を用いて実験を行う．Factorization Machine のツールとして libFM（Version 1.4.2）[7] を用いる．

多項式カーネル SVM，Factorization Machines は開発データでパラメータチューニングを行い，one-vs-all 法を用いて多値分類を行う．また，Factorization Machines のパラメータは相互作用の次元は $d = 2$ で固定し，行列分解したあとの次元数である k や，パラメータ推定の手法などは開発データでパラメータチューニングを行う．

4.4 評価

今回，実験の評価指標として precision と recall と F1 スコアを用いて比較を行う．固有表現以外（O）を除いた4種類のタグを用いて評価する．

4.5 結果

表3が Primadhanty らのデータで実験した結果である．[1] また，表4が固有表現タグごとの結果である．図1がそれぞれのタグごとの precision-recall 曲線である．

今回の実験の結果より，Factorization Machines と先行研究の Primadhanty らの手法とで同程度の結果が得られることが分かった．precision が1ポイント低下するものの，recall が1ポイント改善し，F1 では1ポイントの改善となっている．この結果より Factories

[1] Primadhanty らの実験ではクラスタリング素性や POS タグなどの素性も含まれているが，本研究の実験では使用していない．

表 2: 今回の実験で使用する素性

文脈素性：識別する固有表現候補が含まれる文の固有表現の順番を考慮しない右文脈，左文脈（単語が出現したかしないかのバイナリ素性として用いる）
cap=1, cap=0：固有表現候補の単語の最初の文字が大文字かどうか
all-low=1, all-low=0：固有表現候補の単語が全て小文字かどうか
all-cap1=1, all-cap1=0：固有表現候補の単語が全て大文字かどうか
all-cap2=1, all-cap2=0：固有表現候補の単語が全て大文字でかつ，ピリオドがあるかどうか
num-tokens=1, num-tokens=2, num-tokens>2：固有表現候補の単語が 1 単語で構成されているか，または 2 単語か，それ以上か
dummy：Primadhanty らの実験で必要な文脈や文字種素性がない場合のダミー素性

表 3: Primadhanty らのデータで実験した結果

	precision	recall	F1
対数線形モデル	49.75	44.50	46.75
SVM（多項式カーネル）	53.75	50.67	51.94
対数双線形モデル [5]	**62.03**	53.92	55.88
Factorization Machine	60.93	**55.10**	**57.27**

Machines を用いてスパースなデータでも交互作用を考慮し，未知の固有表現をうまく認識できることが分かった．

5 考察

実験の結果からタグ “ORGANIZATION” の結果が特に向上しているが，たとえば “ORGANIZATION” タグとと “OTHER” タグが付いた事例両方に “Vice-President” という単語が前後の文脈に出現している．このとき，“Vice-President” 以外の前後の文脈も似ているが，提案手法は先行研究と比較して正確に分類できているので，提案手法は比較的スパースな文脈素性と固有表現の文字種素性との組み合わせを考慮して分類が正しくできているのではないかと思われる．そして “LOCATION” の精度があまり高くないが，混同行列を確認したところ，“PERSON” タグに間違える事例が多かったので，本実験で使用した素性では組み合わせを考慮しても 2 つのタグを区別できなかったのではないかと思われる．固有表現候補の接頭辞や接尾辞の情報も今回の実験では入れていないので，そのような素性を増やすことで PERSON タグとの混同が減る効果があると期待できる．また，Primadhanty らの実験ではクラスタリング素性や品詞素性を用いているので，本研究の設定より密な素性が多いが，Factorization Machines はスパースな素性のみでも先行研究と同程度の精度を達成している．

図 2 は Factorization Machines において，開発データを用いて行列分解した次元 k を変化させた時の図である．この図より $k = 5$ の場合に F1 スコアが 57.1 ポイントと一番高い結果となった．$k = 1$ から $k = 5$ までの間は増加し，$k = 8$ で一度スコアが下がるが，k の値が大きくなるほど F1 スコアも増加している．この結果から，このデータでは $k = 5$ の時に行列分解した行列が一番良く潜在的な意味を捉えていると言える．Primadhanty らの対数双線形モデルでは 40 次元で一番高い精度を出しており，Factorization Machines では 10 次元よりも小さい次元で一番高い精度を出している．これは Factorization Machines では行列分解することで小さい次元で潜在的な意味が取れているのに対し，Primadhanty らのモデルでは 40 次元まで潜在的な意味が取りきれないのではないかと思われる．Factorization Machines は少ないパラメータで行列分解することで同程度の精度を達成することができる，という点が優位である．

Primadhanty らの核ノルムを用いた正則化モデルと Factorization Machines はデータスパースネスの解消のために行列の次元削減をしているという点で同じであるが，2 つのモデルでは最適化する目的関数が違っている．核ノルムを用いた正則化モデルでは対数線形モデルに基づいて，SVD で行列分解しているが，Factorization Machines ではヒンジロスを最適化することで直接的に行列分解の最適化ができるという違いが今回の結果を生んだのではないかと思われる．

6 おわりに

この論文では未知の固有表現を認識するタスクにおいて，Factorization Machines を用いて行列分解することでスパースな素性同士の組み合わせを考慮した実験を行った．Factorization Machines を用いることで少ない次元数で先行研究の手法と同程度の精度を得られることが分かった．

今後の課題として，Primadhanty らの実験で使用されていたクラスタリング素性や品詞タグを増やして実験することでスパースネスの解消を検証したい．

表 4: タグごとの固有表現認識の結果

	PERSON			LOCATION			ORGANIZATION			MISC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SVM（多項式カーネル）	**86.45**	72.28	78.73	31.35	38.62	34.61	62.54	**59.40**	60.93	34.67	32.39	33.50
対数双線形モデル [5]	73.83	**90.84**	81.46	**64.96**	36.18	**46.48**	**72.11**	44.98	55.41	37.20	**43.66**	40.17
Factorization Machine	84.36	80.40	**82.33**	39.49	**50.41**	44.29	70.88	55.33	**62.15**	**48.99**	34.27	**40.33**

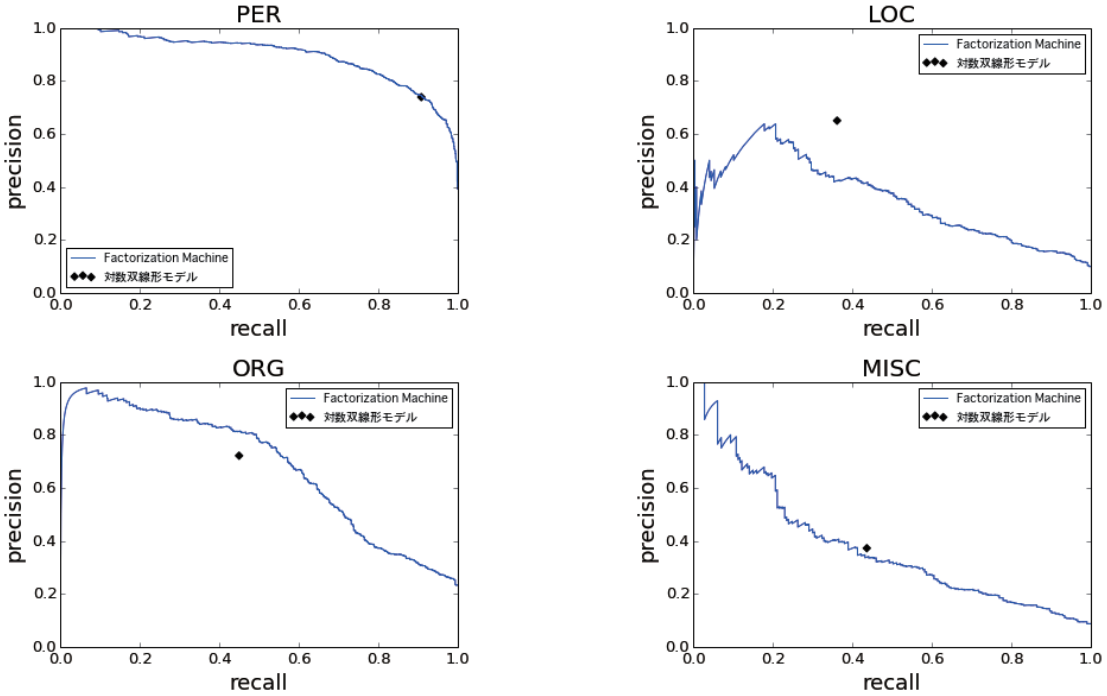

図 1: それぞれのタグにおける precision-recall 曲線

参考文献

[1] 笹野遼平, 黒橋禎夫. 大域的情報を用いた日本語固有表現認識. 情報処理学会論文誌, Vol. 49, No. 11, pp. 3765–3776, 2008.

[2] 福岡健太. Semi-Markov conditional random fields を用いた固有表現抽出に関する研究. 修士論文, 奈良先端科学技術大学院大学情報科学研究科, 2006.

[3] 山田寛康, 工藤拓, 松本裕治. Support vector machine を用いた日本語固有表現抽出. 情報処理学会論文誌, Vol. 43, No. 1, pp. 44–53, 2002.

[4] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *EMNLP*, pp. 100–110, 1999.

[5] Audi Primadhanty and Xavier Carreras Ariadna Quattoni. Low-rank regularization for sparse conjunctive feature spaces: An application to named entity classification. In *Proceedings of ACL-IJCNLP*, pp. 126–135, 2015.

[6] Steffen Rendle. Factorization machines. In *Proceedings of ICDM*, pp. 995–1000, 2010.

[7] Steffen Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, Vol. 3, No. 3, pp. 57:1–57:22, 2012.

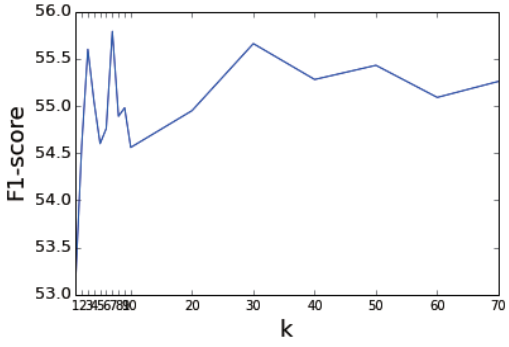

図 2: Factorization Machines における次元数 k を変化させた時の精度