Vol.2017-NL-234 No.5
2017/12/20

教師なし系列マッチング

和田 崇史[1,a)]　岩田 具治[2,b)]

概要：異なるドメイン間のデータで対応関係を見つけることは, 長年取り組まれてきた重要な問題である. しかし, 多くの既存手法では人手で作成されたドメイン間でパラレルなデータを用いて対応関係を学習するため, そのような教師データがない場合には適用できないという問題がある. そこで本稿では, パラレルなデータを用いずに異なるドメインの系列データを対応づける手法を提案する. 全ドメインで共通の LSTM を持つ系列 AutoEncoder を学習することによって, 各ドメインの特徴量を共通の空間で表現する線形写像を獲得し, 異なるドメインで共通した系列のダイナミクスを獲得することを可能にする. 複数言語の対訳文のマッチングにより, 本提案法の有効性を示す.

Unsupervised Sequence Matching

Takashi Wada[1,a)]　Tomoharu Iwata[2,b)]

1. はじめに

異なるドメインのデータをマッチングさせることは, データ間の関係性を明らかにするために重要なタスクであり, これまでにも様々なマッチングの手法が提案されてきた [1–9]. その多くは教師データを元にデータ間の対応関係を学習する手法であり, 例えば 2 つの言語で同じ意味の文をマッチングする場合はパラレルコーパスや対訳辞書を用いて対応関係を学習することが多い. しかし, このようなパラレルな教師データの生成には多くのコストがかかる上, 希少なデータであれば必ずしもそうしたデータが手に入るとは限らないため, そのような教師データなしで対応関係を学習する手法が望まれる. そこで本研究では, 複数の異なるドメインの系列データを教師なしでマッチングさせる手法を提案する. 全ドメインで共通の LSTM [10] を持つ系列 AutoEncoder [11] を学習をすることで, 各ドメイン固有の分散表現が共通の空間上で表さられ, 異なるドメインの系列データのマッチングが可能となる. なお, 本稿では異なる言語で同じ意味の文をマッチングさせることで提案手法の有効性を示すが, 本研究の提案手法はデータの種類に依存せず, 様々な系列データに適用可能な汎用性の高いモデルである.

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[2] NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories
[a)] wada.takashi.wp7@is.naist.jp
[b)] iwata.tomoharu@lab.ntt.co.jp

2. 提案手法

本研究では, 系列データのマッチングを行う. まず, 入力となるのは D 個のドメインの離散系列データ $x_n^d(x_{n,1}^d, x_{n,1}^d, ...x_{n,T_n^d}^d); n = 1, ...N_d, d = 1, ..., D$ である. ここで N_d はドメイン d の全系列数, T_n^d は x_n^d の系列長を示す. なお, 系列データ x_n^d の各要素はドメイン間で異なっており, ドメイン間で共通の要素を抽出して類似度を測ることはできないものとする. ただし, ドメイン間の潜在ダイナミクスは共通していると仮定しており, 各ドメイン間で対応関係にある要素の系列データ中での順序が, それぞれのドメイン間で近いものとする. そして, 各系列 x_n^d からドメイン間の類似度が定義できる実数値ベクトル u_n^d を求め, このコサイン類似度 $\mathrm{cossim}(u_n^d, u_{n'}^{d'})$ を用いて異なるドメインの系列を対応付ける.

本研究では, u_n^d を求めるのにマルチドメイン系列 AutoEncoder という手法を提案する. マルチドメイン系列 AutoEncoder とは, 全ドメインで共通の LSTM を持つ系列 AutoEncoder [11] を学習することによって, 各ドメイン

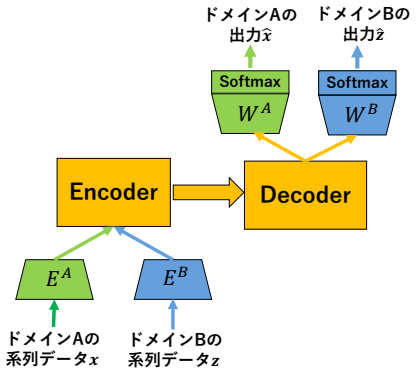

図 1 本研究が提案するマルチドメイン系列 AutoEncoder．ドメイン A の入力 x と出力 x̂, 及びドメイン B の入力 z と出力 ẑ がそれぞれ近くなるように学習を行う．なお, Encoder-Decoder の LSTM のパラメータは全ドメインで共通のものを学習し, 各ドメインの分散表現 E 及び出力の線形写像 W はそれぞれ独立に学習する．

の特徴量を共通の空間で表現する線形写像を獲得し, 異なるドメインで共通した系列のダイナミクスを獲得する手法である (図 1)．

以下, 本提案手法の詳細を示す．系列 AutoEncoder は Encoder と Decoder に分かれており, Encoder では Bi-directional RNN [12] を用いて系列データ x^d をエンコードする．より具体的には, x^d の t 番目の要素 x^d_t の隠れ層 s^d_t は, x^d_t が入力された時の順向き Encoder の隠れ層 $\overrightarrow{s^d_t}$ と逆向き Encoder の隠れ層 $\overleftarrow{s^d_t}$ の和

$$s^d_t = \overrightarrow{s^d_t} + \overleftarrow{s^d_t} \tag{1}$$

で表され, $\overrightarrow{s^d_t}$ 及び $\overleftarrow{s^d_t}$ は以下の式で表される．

$$\overrightarrow{s^d_t} = \mathrm{f_{fwd}}(\overrightarrow{s^d_{t-1}}, \mathrm{E^d x^d_t}) \tag{2}$$

$$\overleftarrow{s^d_t} = \mathrm{f_{bkw}}(\overleftarrow{s^d_{t+1}}, \mathrm{E^d x^d_t}) \tag{3}$$

ここで, $\mathrm{f_{fwd}}$, $\mathrm{f_{bkw}}$ は RNN による非線形変換で, 本研究では Long Short-Term Memory units (LSTM) [10] を用いた．なお, x^d_t は要素の id を表す one-hot ベクトルとなり, E^d の線形写像によって得られる分散表現が LSTM へと入力される．ここで, $\mathrm{f_{fwd}}$ 及び $\mathrm{f_{bkw}}$ は全ドメインでシェアされているが, E^d は各ドメインで独立に学習される．そして, Decoder の隠れ層 h^d_t の初期値 h^d_0 を, 順向き Encoder の最後の要素 x^d_T の隠れ層 $\overrightarrow{s^d_T}$ と, 逆向き Encoder の最初の要素 x^d_0 の隠れ層 $\overleftarrow{s^d_0}$ の和

$$h^d_0 = \overrightarrow{s^d_T} + \overleftarrow{s^d_0} \tag{4}$$

とし, Decoder のセル状態の初期状態も同様に各方向の Encoder の最終セル状態の和とする．そして, これらの情報を元に入力文と同じ文の生成を行うよう学習する．具体的な Decoder の式は以下の通りである．

$$h^d_t = \mathrm{f_{dec}}(\mathrm{h^d_{t-1}}, \mathrm{E^d y^d_{t-1}}) \tag{5}$$

$$p(y^d_t | y^d_{<t}, x^d_t) = \mathrm{softmax}(\mathrm{W^d h^d_t}) \tag{6}$$

ここで, $\mathrm{f_{dec}}$ は LSTM による非線形変換, y^d_t は式 (6) の argmax を取ることで Decoder が t 番目に生成した要素の one-hot ベクトルである．Encoder と同様, $\mathrm{f_{dec}}$ は全ドメインで共通しているが, 行列 W^d 及び E^d は各ドメインごとで独立に学習される行列であり, 式 (5) における E^d は Encoder の式 (2),(3) のものと共通である．なお, 近年の Encoder-Decoder の生成モデル, 特に機械翻訳においてはこの構造に Attention 機構 [13] を適用したモデルが主流であるが, 系列 AutoEncoder では単に Encoder の入力文を Decoder へと Attention 機構でコピーするように学習が進んでしまい, マッチングに必要な系列データの分散表現を Encoder でうまく獲得することができないため, 本研究では注意機構は用いていない．

また, 本研究ではそれぞれの系列データの意味空間がより近くなるように, 次の工夫をモデルに施した．まず, Decoder への最初の入力 y^d_0 (<BOS> トークン) の分散表現を, 各ドメインで同じものを用いた．また, 系列の生成の終了を示す <EOS> トークンが出力される確率を h^d_t から求めるための線形写像も, 各ドメインで同じものを用いた．従って, 式 (6) は以下のように書き換えらる．

$$p(y^d_t | y^d_{<t}, x^d_t) = softmax([W^{\mathrm{EOS}} h^d_t, W^d h^d_t]) \tag{7}$$

ここで, $[x, y]$ は x と y の結合を意味し, W^{EOS} は $1 \times h$ の行列, W^d は $V^d \times h$ の行列, h は h^d_t の次元数, V^d は <EOS> トークンを除くドメイン d の全要素の数である．W^{EOS} は全ドメインで共通のものを学習する．

そして, 以上の手法で全ドメインの系列 AutoEncoder を学習した後, 系列データ x^d を表す実数値ベクトル u^d を次のように求める．

$$u^d = \frac{1}{T} \sum_{t=1}^{T} s^d_t \tag{8}$$

T は x^d の系列長である．すなわち, x^d の各要素の Encoder 隠れ層 s^d_t の平均を系列データの分散表現 u^d とし, そのコサイン類似度を測ることで各ドメインの系列データのマッチングを行う．

3. 関連研究

対応関係を示す教師データなしで異なるドメイン間のオブジェクトを教師なしでマッチングさせる一般的な手法は KS (Kernelized Sorting) [1]，CKS (Convex Kernelized

Sorting) [2], MCCA (Matching CCA) [3]，LSOM (Least-Squares Object Matching) [4], ReMatch (Relataional Matching) [5], MMLVM (Many-to-many Matching Latent Variable Model) [6] などがこれまでに提案されてきた．しかし，これらのオブジェクトマッチングの手法では系列データを適切にマッチングできないという問題があった．また，これらの手法では組合せ最適化問題の一種である線形割当問題を解くことによってマッチングを行うが，それには $O(n^3)$(n はオブジェクトの数) の計算量が必要なため，データの数が増えると膨大な計算時間がかかるという問題もある．

また系列データの中でも特に，異なる言語の対応関係を教師なしで学習するモデルとしては，[7–9] があげられる．これらはまず word2vec [14] や FastText [15] によって単語の分散表現を各言語で独立に得た後，それらを同じ空間へとマップする線形写像を教師なしで学習する手法である．ただし，上記の方法は単語のマッチングを行う手法であり，文のような系列データのマッチングではうまく行かない可能性がある．加えて，これらの手法には以下の問題点が存在する．まず，[7] は実際には各言語におけるアラビア数字の分散表現の対応関係を用いて線形写像を学習しているため，そのような明らかな対応関係がドメイン間で存在していないデータには適用できないという制約がある．また，[7–9] はいずれも，各言語の分散表現の間に線形の関係性があることを前提とした手法であるため，正確な分散表現を得るのに充分な量のデータが各ドメインで存在しない場合には適用できない可能性がある．一方，本提案手法は比較的小規模のデータでも学習可能でかつ非常にシンプルなモデルであり，様々なデータに適用可能な非常に汎用性の高いモデルである．

また，本提案手法では異なるドメインのデータを共通の LSTM に入力してモデルの学習を行ったが，そうした異なるドメイン間でパラメータをシェアするニューラルネットワークのモデルは一般に Siamese Neural Network [16] とよばれており，それぞれのデータを同じ空間で表現することができると知られている．例えば，[17] では本研究と同様，共通の LSTM に 2 つの言語の文を入力し，教師データを用いて異なる言語の文の意味的類似度を測るモデルを提案している．また，[18] では複数の言語の翻訳モデルを全言語共通の Attention-based Encoder-Decoder で学習することに成功しており，異なる言語で同じ意味の文を Encoder に入力した際に，Attention 機構によって得られる context vector の分散表現が異なる言語間で距離が近くなることが確認されている．すなわち，共通の Encoder-Decoder モデルで複数の言語の文の分散表現を学習した場合，それぞれが同じ空間で表現されることを示している．

4. 実験

4.1 評価方法

本研究では，提案手法の有効性を示すために対訳文のマッチングを教師なしで行なった．具体的には，ソース言語の文それぞれに対し，文の分散表現のコサイン類似度が高いターゲット言語の文 Top10, Top5, Top1 を抽出して，それらの中にソース言語の文の対訳文が含まれているかどうかでマッチングの精度を測った．なお，提案手法では文の分散表現は式 (8) で表される．

4.2 データセット

学習及び評価に用いたデータは，Europerl-v7 [19] の英語，スペイン語，イタリア語のパラレルコーパスである．まず，英語–イタリア語と英語 –スペイン語の対訳コーパスから，英語を中間言語として 3 言語間で同じ意味のスペイン語–英語–イタリア語の文をそれぞれ 52000 文ランダムに抽出し，[ソース言語–ターゲット言語] のペアをそれぞれ [スペイン語–英語 (es-en)], [イタリア語–英語 (it-en)], そして [イタリア語–スペイン語 (it-es)] と変えて実験を行なった．ただし，教師なしで各言語のマッチングを行うことが本研究の目的であるため，各コーパスの文のオーダーをシャッフルし，それぞれを独立の単言語コーパスとして用いた．なお，学習，開発，評価に用いた文の数は，各言語で 50000, 1000, 1000 文である．すなわち，1 つの言語ペア毎に 100000 文の学習データを用いてマルチドメイン系列 AutoEncoder を学習する．そして，モデル選択の基準には，開発において 2 言語の AutoEncoder の loss (cross entropy) の合計が最も低かったモデルを最良のモデルとし，文のマッチングに用いた．なお，データの前処理として tokenize, lowercase, cleaninig を Moses [20] のツールキット[*1]を用いて行い，文長が 50 単語を越えない文のみを学習及び評価に使用した．各言語の語彙サイズは英語，スペイン語，イタリア語でそれぞれ 24k, 35k, 35k であった．

4.3 ベースライン

今回，ベースラインとして本提案手法を 4 つのマッチングの手法と比較した．1 つ目は完全にランダムでマッチングを行なった時の正解確率（ランダムマッチング）,2 つ目は系列データの長さが最も近い文をランダムに抽出してマッチングする手法（系列長マッチング）, 3 つ目が異なるドメインで独立に学習された単語の分散表現間の線形写像を，対訳辞書データなしで学習する教師なし単語マッピング [7], そして 4 つ目が Convex Kernelized Sorting [2] である．以下，後半 2 つの手法の概要を説明する．

[*1] https://github.com/moses-smt/mosesDecoder

4.3.1 教師なし単語マッピング

まず学習データを用いて, ソース言語とターゲット言語の単語の分散表現 X, Z を word2vec により独立に学習した後, 各言語のコーパスから正規表現 [0-9]+でマッチングするオブジェクトを抽出し, それらを両言語の初期の対訳辞書として利用する（例: 1-1, 2-2, 1992-1992）. そして, お互いの分散表現の空間が近くなるような直行行列による線形写像 W を以下の式で学習する.

$$W^* = \arg\min_W \sum_i \sum_j D_{ij} ||X_{i*}W - Z_{j*}||^2 \quad s.t. \quad WW^T = W^TW = I \tag{9}$$

X_{i*} はソース言語における i 番目の単語の分散表現を示し, Z_{j*} はターゲット言語における j 番目の単語の分散表現を示す. また, D_{ij} は辞書の対応関係を示し, もしソース言語の単語 i とターゲット言語の単語 j が対訳関係にあれば $D_{ij} = 1$, それ以外は $D_{ij} = 0$ となる. そして, [21] に習い, W を学習する前に X と Z をそれぞれあらかじめベクトル長の正規化及び中心化することにより, 式 (9) は以下の式で効率的に求められる.

$$W^* = \arg\max_W Tr(XWZ^TD^T) \tag{10}$$

$Tr(\cdot)$ はトレース演算子を示す. 式 (10) で線形写像 W を学習したのち, 以下のように対訳辞書 D を更新する.

$$D_{ij} = \begin{cases} 1 & \text{if} \quad j = \arg\max_k (X_{i*}W) \cdot Z_{k*} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

そして, 更新された対訳関係 D を用いて, 再び式 (10) で新たな W を得る. この self-learning のステップをアルゴリズムが収束するまで繰り返し, 最後に得られた W を両言語の線形写像とする. すなわち, アラビア数字のみの対訳辞書でおおよその線形写像 W を学習した後, 写像後の空間において距離が一番近い単語同士を対訳関係にあるとみなして, より正確な線形写像 W を段階的に学習する手法である.

なおイタリア語–英語のマッチングでは, アラビア数字の対訳関係の代わりに対訳辞書データ D を与えて式 (10) で線形写像を学習する教師ありの手法も本提案手法と比較した. また, 本手法の実装と英語–イタリア語の対訳辞書は, [7] の著者によって公開されているコード及びデータを用いた[*2]. 英語–イタリア語の対訳辞書は, Europerl のパラレルコーパスにおいて高頻度で現れた単語のアラインメントによって作成されたものである.

そして文の分散表現としては, 文中の未知語を除いた単語の写像後の分散表現の平均を用いた. したがって, ベースラインにおいては提案手法とは異なり, 文の分散表現に語の語順が考慮されていない, いわゆる bag of words の形で表されていることとなる.

[*2] https://github.com/artetxem/vecmap

,

	es-en	it-en	it-es
ランダムマッチング	0.001	0.001	0.001
系列長マッチング	0.000	0.000	0.002
教師なし単語マッピング	0.137	0.047	0.246
CKS	0.046	0.021	0.019
提案手法	0.265	0.185	0.397
提案手法+ <BOS>, <BOS> の共有	**0.344**	**0.191**	**0.504**

表 1 各言語対において, 最もコサイン類似度が高い文でマッチングした時のマッチングの正解率

	Top1	Top5	Top10
ランダムマッチング	0.001	0.005	0.010
系列長マッチング	0.002	0.015	0.059
教師なし単語マッピング	0.246	0.470	0.568
CKS	0.019	–	–
提案手法	0.397	0.590	0.682
提案手法+ <BOS>, <BOS> の共有	**0.504**	**0.699**	**0.759**

表 2 イタリア語–スペイン語における, 各手法のマッチングの正解率. Top1, Top5, Top10 はそれぞれコサイン類似度が最も高い文を 1, 5, 10 文抽出した時に正解が含まれている率を示す

4.3.2 Convex Kernelized Sorting

Convex Kernelized Sorting (CKS) [2] とは, を 2 つの異なるドメイン間の依存関係を最大化させるようにデータを並べ換え, 両者を教師なしでマッチングさせる手法である. ここで依存関係を測るのに用いられるのはヒルベルト-シュミット独立基準（HSIC）[22] である. 本研究では提案手法との比較のためテストデータ 1000 文を CKS を用いてマッチングを行なった. CKS が他の手法と大きく異なるのは, CKS は学習データを一切用いず, 評価データのみからマッチングを行う点である. また, 本手法は 2 つのドメイン間のデータを一対一対応させる手法であるため, Top1 の結果のみを比較する. なお, 各文の特徴量を文中に現れる各単語の頻度を要素とする bag of words で表されたベクトルで表現した. また, 実装には [2] の著者によって公開されているコードを使用した[*3].

4.4 学習

2 つの言語でミニバッチを独立に作り, 1 ミニバッチごとに言語を変えて各言語の AutoEnocder の cross entropy が下がるようにモデルの最適化を行った. 最適化には SGD [23] を用いて, 学習率の初期値を 1.0 とし, 開発において 2 言語の AutoEncoder の loss の合計が上がった時に学習率を 0.7 倍することで, 学習率を徐々に減衰させた. エポック数は 60, バッチサイズは 64, Encoder と Decoder の隠れ層の次元数及び単語の分散表現の次元数は 500 とした. また式 (6) 及び (7) において, 過学習を防ぐためにドロッ

[*3] https://astro.temple.edu/~tuc17157/codes/CKS_code.zip

プアウト [24] を Decoder の隠れ層 h_t^d にドロップアウト率 0.3 で適用した．モデルのパラメータの初期値は [-0.1, 0.1] の一様分布から得るランダムの値を用いた．

5. 結果

表 1 に各言語のマッチングの結果を示す．まず,2 つの本提案手法を比較すると，<BOS> トークン及び <EOS> トークンを各言語間でシェアすることによって，スペイン語–英語のマッチングの精度が+0.079，イタリア語–英語のペアで+0.006，そしてイタリア語–スペイン語のマッチングで+0.107 といずれの言語対でもマッチングの精度が向上していることがわかる．従って，トークンをシェアすることで各言語の文の分散表現がお互いにより近い空間で表現されていることがわかる．また，教師なし単語マッピングとの比較では，英語–スペイン語のマッチングの精度が+0.207，イタリア語–英語のペアで+0.144，そしてイタリア語–スペイン語のマッチングで+0.258 と全ての言語対において提案手法がベースラインの手法を大幅に上回っていることがわかり，提案手法の有効性が示された．

また，表 2 が示しているのはイタリア語の各文とコサイン類似度が最も高いスペイン語文を上から 10, 5, 1 文それぞれ抽出した時に対訳文が含まれている率である．ここでも，<EOS> トークンと <BOS> トークンをシェアした提案手法が最も高い精度でマッチングができていることが確認できる．

そして，図 2，3 は，イタリア語–英語の単語マッピングを n 個の単語ペアから成る対訳辞書 D($n = 0, 300, 600, 900, 1200, 1500$) を用いて学習した時の Top1 及び Top10 の結果を本提案手法と比較した図である．この時，線形写像は与えられた対訳辞書 D から式 (10) により求めており，式 (11) で対訳辞書の更新を段階的に行う self-learning は行なっていない[*4]．ただし，$n = 0$ の場合は 4.3.1 で示したようにアラビア数字の対応関係を初期の対訳辞書として用い，その後 self-learning を行なった結果である．本提案手法のマッチングは，対訳辞書で 300 または 600 の単語ペアを教師データで与えた手法よりも精度が高く，900 の単語のペアを与えた時と同じくらいの精度であることがわかる．従って，本提案手法は非常に精度よくマッチングができていることがわかる．

6. 分析

表 1 の言語間の結果を比較すると，本提案手法のイタリア語–スペイン語のマッチングの結果が，他の言語対と比べて精度がかなり良いことがわかる．これは，イタリア語とスペイン語の語順や語彙などの言語構造が比較的似ているため，マルチドメイン系列 AutoEncoder で両言語共通のダイナミクスを効率よく学習できたためだと考えられる．特に，提案手法においてはイタリア語–スペイン語の 1000 文のパラレルコーパスのうち 50.4%の文を教師なしでマッチングすることに成功しており，系列データの特徴が近いデータであれば異なるドメインでも高い精度でマッチングを行うことができることが示された．

[*4] 対訳辞書データ D を初期の辞書として self-learning を行うことも試したが，行わない場合に比べて精度が大幅に悪化した．これは，本研究で word2vec の学習に用いたデータが 5 万文と小規模なため，低頻度で現れる単語の分散表現を言語間で線形写像を行うことが難しいためだと考えられる．

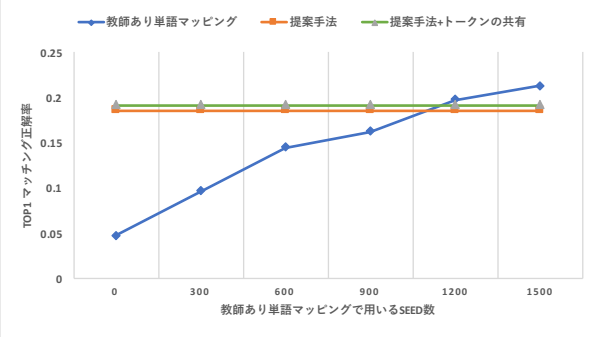

図 2 イタリア語–英語のマッチングにおいて，対訳辞書の単語のペア数 (seed) を変えて単語マッピングを学習した時の，Top1 マッチングの正解率の推移と，本提案手法の精度の比較

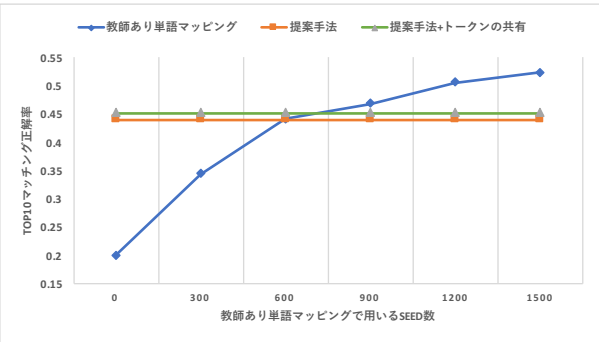

図 3 イタリア語–英語のマッチングにおいて，対訳辞書の単語のペア数 (seed) を変えて単語マッピングを学習した時の，Top10 マッチングの正解率の推移と，本提案手法の精度の比較

7. おわりに

本研究では，マルチドメインで共通の系列 AutoEncoder を学習することで，パラレルコーパスなどの教師データが存在しない場合でも異なるドメインの系列データをマッチングすることができることを示した．そして，マッチングさせる言語のペアを変えて比較することで，系列データの特徴が似ているイタリア語–スペイン語の方がその他の言語対よりもマッチング精度が高いことが明らかとなった．今後，言語以外の系列データに本提案手法を適用したい．

参考文献

[1] Quadrianto, N., Song, L. and Smola, A. J.: Kernelized Sorting, *Advances in Neural Information Processing Systems 21* (Koller, D., Schuurmans, D., Bengio, Y. and Bottou, L., eds.), Curran Associates, Inc., pp. 1289–

1296 (2009).

[2] Djuric, N., Grbovic, M. and Vucetic, S.: Convex Kernelized Sorting, *AAAI Conference on Artificial Intelligence (AAAI)* (2012).

[3] Haghighi, A., Liang, P., Berg-Kirkpatrick, T. and Klein, D.: Learning Bilingual Lexicons from Monolingual Corpora, *Proceedings of ACL-08: HLT*, Columbus, Ohio, Association for Computational Linguistics, pp. 771–779 (2008).

[4] Yamada, M. and Sugiyama, M.: Cross-Domain Object Matching with Model Selection, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)* (Gordon, G. J. and Dunson, D. B., eds.), Vol. 15, Journal of Machine Learning Research - Workshop and Conference Proceedings, pp. 807–815 (2011).

[5] Iwata, T., Lloyd, J. R. and Ghahramani, Z.: Unsupervised Many-to-Many Object Matching for Relational Data, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 38, No. 3, pp. 607–617 (2016).

[6] Iwata, T., Hirao, T. and Ueda, N.: Unsupervised Cluster Matching via Probabilistic Latent Variable Models, *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, AAAI Press, pp. 445–451 (2013).

[7] Artetxe, M., Labaka, G. and Agirre, E.: Learning bilingual word embeddings with (almost) no bilingual data, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, Association for Computational Linguistics, pp. 451–462 (2017).

[8] Zhang, M., Liu, Y., Luan, H. and Sun, M.: Adversarial Training for Unsupervised Bilingual Lexicon Induction, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, Association for Computational Linguistics, pp. 1959–1970 (2017).

[9] Conneau, A., Lample, G., Ranzato, M., Denoyer, L. and Jégou, H.: Word Translation Without Parallel Data, *CoRR*, Vol. abs/1710.04087 (2017).

[10] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780 (1997).

[11] Dai, A. M. and Le, Q. V.: Semi-supervised Sequence Learning, *CoRR*, Vol. abs/1511.01432 (2015).

[12] Schuster, M. and Paliwal, K.: Bidirectional Recurrent Neural Networks, *Trans. Sig. Proc.*, Vol. 45, No. 11, pp. 2673–2681 (1997).

[13] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *CoRR*, Vol. abs/1409.0473 (2014).

[14] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *CoRR*, Vol. abs/1301.3781 (2013).

[15] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *arXiv preprint arXiv:1607.04606* (2016).

[16] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R.: Signature Verification Using a "Siamese" Time Delay Neural Network, *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp. 737–744 (1993).

[17] Neculoiu, P., Versteegh, M. and Rotaru, M.: Learning Text Similarity with Siamese Recurrent Networks, *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany, Association for Computational Linguistics, pp. 148–157 (2016).

[18] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M. and Dean, J.: Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, *CoRR*, Vol. abs/1611.04558 (2016).

[19] Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation, *Conference Proceedings: the tenth Machine Translation Summit*, Phuket, Thailand, AAMT, AAMT, pp. 79–86 (2005).

[20] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 177–180 (2007).

[21] Artetxe, M., Labaka, G. and Agirre, E.: Learning principled bilingual mappings of word embeddings while preserving monolingual invariance, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, Association for Computational Linguistics, pp. 2289–2294 (2016).

[22] Gretton, A., Bousquet, O., Smola, A. and Schölkopf, B.: Measuring Statistical Dependence with Hilbert-schmidt Norms, *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, ALT'05, Berlin, Heidelberg, Springer-Verlag, pp. 63–77 (2005).

[23] Bottou, L.: Large-Scale Machine Learning with Stochastic Gradient Descent, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)* (Lechevallier, Y. and Saporta, G., eds.), Paris, France, Springer, pp. 177–187 (2010).

[24] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958 (2014).