単語埋め込みを利用した商品に対するキーワードの予測

山口 泰弘　深澤 祐援　原島 純
クックパッド株式会社
{ yasuhiro-yamaguchi, yusuke-fukasawa, jun-harashima }@cookpad.com

概要

出品者が商品情報を自由に入力できる EC サービスでは，入力される商品名の多様性のため，出品された商品がなにであるかを機械的に判断することは難しい．本研究では，事前学習した単語埋め込みを用いて，予め定義されたキーワード集合の中から商品名に最も関連するものを予測するモデルを提案する．クックパッドマートの商品を元に作成したデータセットによる実験の結果，提案手法では Accuracy@5 で 95.8% の精度でキーワードの予測が可能になった．また，注意機構を用いることで教師なしの手法においても Accuracy@5 で 81.0% の精度で予測できることがわかった．

1　はじめに

出品者が商品情報を自由に入力できる EC サービスでは商品名や商品説明の表記は出品者によって大きく異なる．例えば食材を多く扱う EC サービスであるクックパッドマートでは，豚の小間切れ肉の商品名について “豚肉小間切れ”，“豚コマ”，“豚小間肉” など複数の表記が存在する．さらに，じゃがいもなどの野菜では “メークイン” や “キタアカリ” といった品種名が記載される場合も多い．また，ユーザーの購買意欲を高めるために “産地直送” や “タイムセール” といったフレーズを含んでいるものもある．検索や推薦といったタスクでは商品のカテゴリや性質は有用な情報になり得るが，こうした表記の多様性はその商品のカテゴリを機械的に判別することを困難にする．

そこで本研究では，事前学習した単語埋め込みに基づく商品名とキーワードの類似性を利用してキーワード集合の中から商品に関連するキーワードを割り当てる機械学習モデルを提案する．表 1 に本研究で利用するデータセットの例を示す．データセットはクックパッドマートの商品をもとに作成した商品名とキーワードのペアからなり，モデルは与えられた商品名から対応するキーワードを予測する．本研究ではこのデータセットに基づいて提案手法の有効性を検証する．

商品名	キーワード
産地直送キュウリ 2 本入れ	きゅうり
天然白ミル貝（生食用/殻付き）	ミル貝
【タイムセール】カクテルスイートペッパー	パプリカ
自社農園産 キタアカリ	じゃがいも
宮城県産えごま豚 小間切れ	豚肉 小間切れ
北海道産　朝どれグリーンアスパラ	アスパラガス

表 1　データセットの例

2　関連研究

EC サービス上の商品名を元に商品をカテゴリに分類する手法がいくつか提案されている．Cevahir ら [1] は商品を階層化された 2 万以上のカテゴリに分類するために自己符号化器の近傍などを利用して階層的に予測を行う分類モデルを提案した．

Chen ら [2] は商品を詳細なカテゴリに分類するために，商品名や説明文中の商品カテゴリに相当する語に着目するニューラル分類器を提案している．

Cevahir らと Chen らの手法はいずれもカテゴリを分類ラベルとして予測するものであり，商品情報とカテゴリの文字列的・意味的な類似性を直接的に考慮するものではないという点で本研究とは異なっている．

また，テキストデータから商品の情報を抽出する先行事例として [3] や [4] がある．これらの手法は系列ラベリングを用いてテキスト中から属性情報を抽出する．本研究は予め定義されたキーワード集合に分類するという点でこれらの手法とは異なる．

3　提案手法

本研究では，事前に定義したキーワード集合の中から与えられた商品名に最も関連のあるキーワードを予測するモデルを提案する．

図 1 に提案モデルの構造を示す．このモデルでは商品名とキーワードの類似度を計算し，商品名との

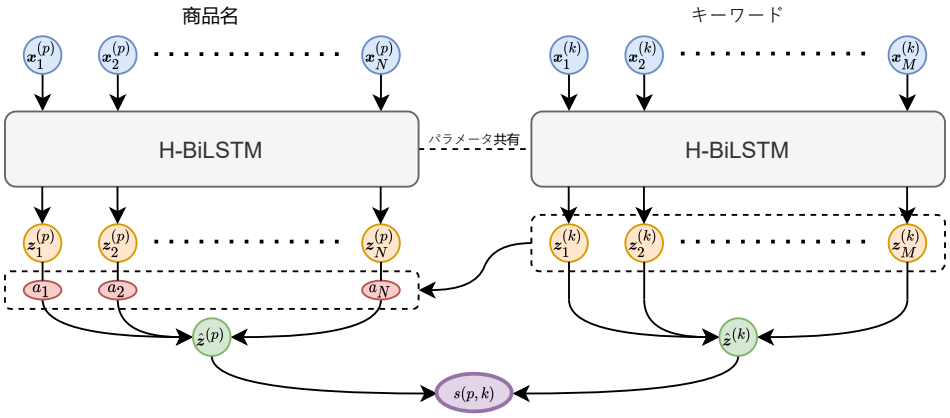

図 1　キーワード予測モデルの構造

類似度が最も大きいキーワードを選択することで予測を行う．商品名とキーワードの類似度の計算手順を次に示す:

1. 商品名・キーワードに対して形態素解析を行い単語ごとに分割する
2. 商品名・キーワードの各単語を事前に学習した fastText[5] を用いてベクトルに変換する
3. 商品名・キーワードの単語ベクトル列をそれぞれ Highway BiLSTM により変換する (3.1 節)
4. 3 で変換したベクトルをもとに注意機構を用いて商品名の単語ベクトルをキーワードの単語ベクトルとの類似度で重み付ける (3.2 節)
5. 商品名・キーワードの単語ベクトル列をそれぞれ平均して商品名ベクトル・キーワードベクトルを計算する (3.2 節)
6. 商品名ベクトル・キーワードベクトル同士のコサイン類似度を商品名とキーワードの類似度とする (3.3 節)

3.1　Highway BiLSTM (H-BiLSTM)

本研究では学習済み fastText の特徴量を効果的に利用するために Highway 構造 [6] を導入した BiLSTM をエンコーダーに利用する．ここでは Hightway BiLSTM (H-BiLSTM) を以下のように定義する．$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T)$ は BiLSTM の入力となるベクトル系列であり，σ はシグモイド関数を表す．

$$\boldsymbol{H} = \mathrm{BiLSTM}_{\mathrm{highway}}(\boldsymbol{X}) \tag{1}$$

$$\boldsymbol{G} = \sigma(\mathrm{BiLSTM}_{\mathrm{gate}}(\boldsymbol{X})) \tag{2}$$

$$\boldsymbol{Z} = \boldsymbol{G} \circ \boldsymbol{H} + (1 - \boldsymbol{G}) \circ \boldsymbol{X} \tag{3}$$

事前実験の結果から，本研究では商品名・キーワードの fastText ベクトル列をパラメータを共有した同一の H-BiLSTM を用いてエンコードする．

3.2　注意機構

商品名には “産地直送” や “タイムセール” といったキーワードに直接関係のない語彙が含まれているものが多く存在する．本研究ではこうしたキーワードに無関係な単語の影響を小さくするために，注意機構を用いた商品名の単語に対する重み付けの仕組みを導入する．

商品名 p の n 番目の単語に対する重み a_n をキーワード k の単語とのコサイン類似度を利用して次のように定義する．

$$a_n = \frac{1}{M} \sum_{m=1}^{M} \cos\left(\boldsymbol{z}_n^{(p)}, \boldsymbol{z}_m^{(k)}\right) \tag{4}$$

ここで $\boldsymbol{Z}^{(p)} = \left(\boldsymbol{z}_1^{(p)}, \boldsymbol{z}_2^{(p)}, \cdots, \boldsymbol{z}_N^{(p)}\right)$, $\boldsymbol{Z}^{(k)} = \left(\boldsymbol{z}_1^{(k)}, \boldsymbol{z}_2^{(k)}, \cdots, \boldsymbol{z}_M^{(k)}\right)$ はそれぞれ H-BiLSTM を通して計算した商品・キーワードの単語ベクトル系列とする．

商品名とキーワードに対応するベクトルをそれぞれ以下の定義に従って計算する．

$$\hat{\boldsymbol{z}}^{(p)} = \frac{1}{N} \sum_{n=1}^{N} a_n \boldsymbol{z}_n^{(p)} \tag{5}$$

$$\hat{\boldsymbol{z}}^{(k)} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{z}_m^{(k)} \tag{6}$$

キーワードの単語列は十分短く，予測に不必要な単語が含まれることもほとんどないため単語の重み付けは商品名のみを対象とした．

3.3 類似度と損失関数

商品名 p とキーワード k の類似度 $s(p,k)$ は $\hat{z}^{(p)}$, $\hat{z}^{(k)}$ のコサイン類似度で以下のように定義する．

$$s(p,k) = \cos\left(\hat{z}^{(p)}, \hat{z}^{(k)}\right) \tag{7}$$

モデルの学習では商品名と関連するキーワードのペアからなるデータセット $\mathcal{D}$ を用いて以下の損失関数を最小化する．

$$L = -\sum_{(p,k)\in\mathcal{D}} \log\left(\frac{e^{s(p,k)}}{e^{s(p,k)} + \sum_{k'\in\mathcal{N}_k} e^{s(p,k')}}\right) \tag{8}$$

$\mathcal{N}_k \subseteq \mathcal{K} \setminus \{k\}$ は全キーワード集合 $\mathcal{K}$ から商品名 p の正例であるキーワード k を除いてネガティブサンプリングを行い得られた負例のキーワード集合を表す．対照学習では負例サンプリングにおいて識別困難な事例を優先的に利用することでモデルの性能向上につながることが知られている [7]．本研究では，1 つの事例に対して負例のキーワード集合の中から 128 件をランダムに選んで商品名との類似度を計算し，この中から類似度が最も小さい上位 8 件のキーワードを hard example として損失の計算に利用した．

4 実験

4.1 データセット

クックパッドマートに出品されている 8,881 件の商品に対して食材を表すキーワードの割り当てを人手で行った．一度アノテーションしたあとで割り当てたキーワードに対する表記揺れの修正やキーワードの統合などの処理をした結果，ユニークなキーワードの数は 1,350 件となった．アノテーションしたデータセットをランダムに Train, Valid, Test の 3 つに分割し，モデル学習と評価に利用した．データセットのより詳細な統計情報は付録 A に記す．

4.2 比較手法

本研究では以下のモデルを用いて性能の比較を行った．

TF-IDF 類似度: 商品名・キーワードをそれぞれ単語分割した後 TF-IDF によるベクトル化を行い，TF-IDF ベクトルのコサイン類似度が最も大きい商品名・キーワードを対応づける教師なしモデル．

fastText 分類器: 商品名の fastText ベクトル列を平均した商品名ベクトルを入力とし，順伝播型ニューラルネットワークを通してキーワードごとにクラス分類を行う教師ありモデル．

fastText 類似度: 商品名・キーワードの単語ベクトル列の平均ベクトルを計算し，各ベクトルのコサイン類似度が最も大きい商品名・キーワードを対応づける教師なしモデル．

fastText 類似度 + 注意機構: 商品名の fastText ベクトルを注意機構で重み付けした後，fastText 類似度を計算する手法．提案手法から H-BiLSTM 層を除いたモデルと同等．

H-BiLSTM 類似度: 提案手法から注意機構を除いた教師ありモデル．

H-BiLSTM 類似度 + 注意機構: 本研究の提案手法である教師ありモデル．

本研究では単語埋め込みとしてクックパッドに公開されている約 350 万レシピのタイトルを用いて学習した fastText モデルを利用した．fastText の詳細は付録 B に記す．レシピのタイトルの単語分割には MeCab (Unidic 辞書) を用いた．

各モデルは Train データで学習し，Valid データでハイパーパラメータの選択と Early Stopping を行い，Test データを用いて精度の比較を行った．

5 結果と考察

	Accuracy@1	Accuracy@5	MRR
TF-IDF 類似度	40.7	65.6	51.3
fastText 分類器	53.8	75.2	63.0
fastText 類似度	50.8	70.4	60.1
+ 注意機構	62.8	81.0	71.1
H-BiLSTM 類似度	77.8	94.8	85.1
+ 注意機構	**78.7**	**95.8**	**86.2**

表 2 Test データに対する実験結果

表 2 に Test データに対する各手法の精度を示す．実験の結果，いずれの指標においても提案手法が最も高い精度となった．

また教師ありの fastText 分類器と教師なしの fastText 類似度+注意機構の精度を比較すると，いずれの指標でも教師なし注意機構モデルが高い精度となった．この結果から，本タスクにおいては単にキーワードをラベルとして予測するよりも注意機構を用いて商品名とキーワードの類似性を考慮した予測を行うことが重要であると考えられる．

商品名	キーワード	予測結果 (上位 5 件)
愛知県産　旨味たっぷりキャベツ	キャベツ	キャベツ, サボイキャベツ, ロールキャベツ, レッドキャベツ
シルクスイート	さつまいも	さつまいも, 安納芋, エバミルク, 金時にんじん, 金時草
ギンダラ　フィレ 1/4 切れ（解凍　天然）	銀だら	銀だら, ヒレ, 牛ヒレ, 銀ムツ, 豚ヒレ肉
無添加 あじ開き 2 枚 (解凍)	アジの開き	アジの開き, アジの干物, アジ, 刺身用アジ, サバ開き
お刺身用生桜海老（台湾産）	サクラエビ	刺身, サクラエビ, むきえび, 甘エビ, えび
紫にんじん	紫にんじん	紫にんじん, 紫キャベツ, 紫大根, 紫アスパラ, 紫白菜

表 3　上位 5 件の予測結果の例

5.1　エラー分析

表 3 にモデルの予測結果の例を示す．“ギンダラ/銀だら” や “シルクスイート/さつまいも” のように表記揺れや品種名の記載によってキーワードの文字列が商品名に含まれない場合であっても関連するキーワードを予測できていることがわかる．

一方，“お刺身用生桜海老（台湾産）” の例では “刺身” が最も関連するキーワードとして予測されている．また，“紫にんじん” に対する予測結果をみてみると “紫” が含まれるキーワードが多く予測されていることがわかる．fastText は単語の共起性に基づいて学習を行うため，“紫” や “刺身” のように多くのコンテキストで出現する語は他の多くの単語に対して高い類似度になる傾向がある．この傾向が提案モデルの予測結果にも現れていると考えられる．

5.2　注意機構の効果

表 2 から fastText 類似度・H-BiLSTM 類似度のそれぞれで注意機構の有無による精度を比較すると，いずれの指標においても注意機構を用いた場合に精度が向上した．特に教師なしの fastText 類似度モデルでは注意機構の利用で各指標がそれぞれ +10% 程度の上昇が確認できた．一方，教師ありの H-BiLSTM 類似度モデルでは fastText 類似度モデルの場合ほど大きな性能向上は見られなかったが，各指標において +1% 程度の改善が見られた．以上の結果から類似度の計算において注意機構を用いることがキーワード予測モデルの性能向上につながると考えられる．

5.3　Highway 構造と Hard Negative Sampling

提案手法における Highway 構造と Hard Negative Sampling の影響を調べるために，Highway BiLSTM を通常の BiLSTM に置き換えたものと，キーワードの Negative Sampling をランダムに行ったものでそれぞれ実験を行った．図 2 に学習中の Valid データに対する Accuracy@5 の推移を示す．

まず Hard Negative Sampling を行った手法で BiLSTM (BiLSTM with HNS) と Highway (H-BiLSTM with HNS) を比較すると，H-BiLSTM の方が高い精度を達成したものの大きな差は見られなかった．一方 HNS を行わずランダムに負例を選択した手法を比較すると，BiLSTM (BiLSTM with NS) では学習を進めると Valid データにおける精度が下がる傾向が見られた．

fastText 類似度+注意機構の教師なしモデルでも比較的高い精度で予測できることから，学習済み fastText が類似度を測る上で重要な特徴量であることがわかる．Highway 構造により学習済み fastText のベクトルをバイパス可能にすることで，fastText の特徴量を効果的に学習に活用できると考えられる．

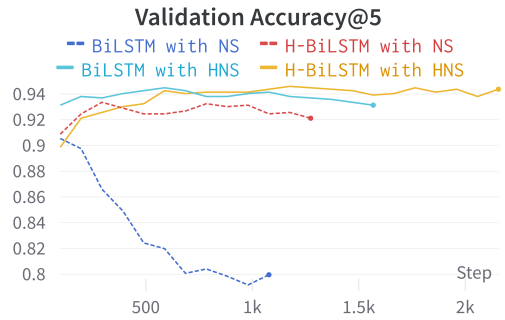

図 2　学習中の Valid データに対する正解率の推移

6　おわりに

本研究では学習済み fastText ベクトルを用いて商品名からキーワードを予測する機械学習モデルを提案した．クックパッドマートのデータを用いた実験の結果，提案手法を用いることで Accuracy@5 95.8% の精度で予測することができた．また，注意機構を用いることで教師なしでも比較的高い精度でキーワードの予測が可能になることが示された．

一方，商品名に出現頻度の高い語が含まれる場合には上位の予測結果として関連の低いキーワードが予測されることが確認できた．より高い精度で予測を行うにはこうした単語埋め込みの特性を考慮した仕組みが必要になると考えられる．

参考文献

[1] Ali Cevahir and Koji Murakami. Large-scale multi-class and hierarchical product categorization for an E-commerce giant. In **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers**, pp. 525–535, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[2] Hongshen Chen, Jiashu Zhao, and Dawei Yin. Fine-Grained Product Categorization in E-commerce. In **Proceedings of the 28th ACM International Conference on Information and Knowledge Management**, pp. 2349–2352, Beijing China, November 2019. ACM.

[3] Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. Text mining for product attribute extraction. **ACM SIGKDD Explorations Newsletter**, Vol. 8, No. 1, pp. 41–48, June 2006.

[4] Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. Scaling up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5214–5223, Florence, Italy, 2019. Association for Computational Linguistics.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, Vol. 5, pp. 135–146, 2017.

[6] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway Networks. **arXiv:1505.00387 [cs]**, November 2015. arXiv: 1505.00387.

[7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 815–823, 2015.

A　データセット統計

本研究で作成したデータセットの統計情報を表 4 に示す．ユニークキーワード数は各サブセットにおけるユニークなキーワードの個数を表す．

	Train	Valid	Test
商品数	6216	887	1775
ユニークキーワード数	1240	484	713

表 4　データセットの統計

各キーワードが割り当てられた商品数の頻度分布を図 3 に示す．この図からキーワードの頻度分布はロングテールであることがわかる．

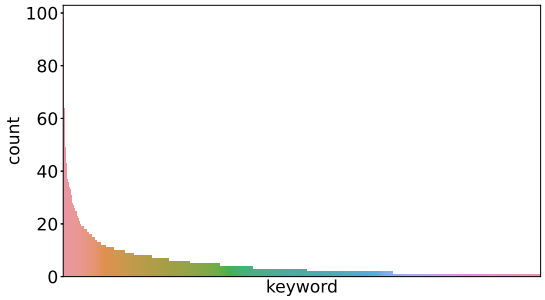

図 3　データセット中のキーワードの頻度分布

B　fastText の詳細

fastText の学習ではクックパッド上のレシピタイトルに対して skip-gram による学習を行なった．ハイパーパラメータの設定は Bojanowski ら [5] の実装における既定値[1)]を利用し，100 次元の単語埋め込みを学習した．

単語	近傍の単語
豚コマ	豚こま, 豚肉, 小間, 豚バラ, 小間切れ
桜エビ	桜えび, 桜海老, アミエビ, 干しエビ, 小えび
さつまいも	さつまいも, 安納芋, 人参芋, 男爵いも, さつま揚げ
りんご	リンゴ, 林檎, 紅玉, アップル, 王林
紫	アーリー, ラディッシュ, 酢漬け, パプリカ, タマネギ
刺身	さしみ, 造り, メバチ, シマアジ, 鮪

表 5　学習した fastText における近傍の単語

学習した fastText ベクトルのコサイン類似度を元に，いくつかの単語の近傍を抽出した結果を表 5 に示す．この表から，fastText の学習によって表記の違いや食材と品種名の対応がある程度考慮できていることがわかる．一方 “紫” や “刺身” といった単語では，同じコンテキストで使われやすい複数種類の食材などが近傍にあることがわかった．

1)　https://github.com/facebookresearch/fastText/tree/v0.9.2/python#train_unsupervised-parameters