INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Character-based Embedding Models and Reranking Strategies for Understanding Natural Language Meal Descriptions

Mandy Korpusik, Zachary Collins, and James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{korpusik, zcollins, glass}@mit.edu

Abstract

Character-based embedding models provide robustness for handling misspellings and typos in natural language. In this paper, we explore convolutional neural network based embedding models for handling out-of-vocabulary words in a meal description food ranking task. We demonstrate that character-based models combined with a standard word-based model improves the top-5 recall of USDA database food items from 26.3% to 30.3% on a test set of all USDA foods with typos simulated in 10% of the data. We also propose a new reranking strategy for predicting the top USDA food matches given a meal description, which significantly outperforms our prior method of n-best decoding with a finite state transducer, improving the top-5 recall on the all USDA foods task from 20.7% to 63.8%.

Index Terms: Convolutional Neural Networks, Crowdsourcing, Character-based embeddings, Levenshtein Edit Distance

1. Introduction

The motivation for our work in the nutrition domain stems from the rising obesity rate in the United States. Adult obesity increased from 13% to 32% between the 1960s and 2004 [1], and today more than one-third of American adults (i.e., 78.6 million) are obese [2], leading to an estimated medical cost of $147 billion in 2008 [3]. Although food journaling is a useful tool for weight loss, existing diet tracking applications such as MyFitnessPal [4] are too time-consuming for many users, requiring manually entering each eaten food by hand and selecting the correct item from a long list of entries in the nutrient database.

Our goal is to provide a simpler diet tracking option for obesity patients by applying speech and language understanding technology to automatically detect food entities and find the corresponding nutrition facts in a database. In our prior work [5], we explored a convolutional neural network (CNN) for directly mapping between users' meal descriptions and the corresponding food database matches. We showed that this method outperformed our previous approach of first tagging foods and properties with a conditional random field (CRF), followed by food-property association, and subsequently using heuristics to map foods to the nutrient database [6, 7].

In this work, we address the problem of what happens when a user misspells a food or brand in their meal description, or when there is a new food that the system has not seen before in the training data. To handle these cases, we develop a character-based CNN that learns embeddings for each character in a token, rather than only at the word level. Thus, with a character model, out-of-vocabulary (OOV) words are represented as character sequences and can be used to predict matching foods with similar characters, while the previous word-based approach would not be able to handle such OOV words.

This research was sponsored by a grant from Quanta Computing, Inc., and by the Department of Defense (DoD) through the National Defense Science Engineering Graduate Fellowship (NDSEG) Program.

2. Related Work

Many researchers in the natural language processing community are currently exploring convolutional neural networks (CNNs) for processing text. In question answering, recent work has shown improvements using deep CNN models for text classification [8, 9, 10], following the success of deep CNNs for computer vision. In other work, parallel CNNs predict the similarity of two input sentences by computing a word similarity matrix between the two sentences as input to a CNN [11, 12, 13]. Attention-based CNNs were also proposed for sentence matching [14] and machine comprehension [15].

Much work has also been done with character-based models. Character-based long short-term memory networks (LSTMs) have been used in neural machine translation for handling OOV words [16], and Google used sub-word units (called "wordpieces"), that performed better than character-based or word-based models for translation [17]. Character-based and word-based embeddings have been combined into joint embeddings for state-of-the-art part-of-speech tagging, which requires syntactic information [18]. Most similar to our approach, Kim et al. learned character-based word embeddings using a CNN followed by a highway network [19]; however, while their task is language modeling, to which they apply a recurrent neural network, ours is binary classification of meal descriptions and USDA food items, to which we apply a CNN. Other related work explored deep character-based CNNs for large-scale text classification [20].

3. Models

In this section, we discuss the model we employed for mapping meal descriptions to USDA food items. There are two steps required to select the top-5 USDA food database matches for a given meal description: 1) train a neural network model that learns vector representations for USDA food items through a binary verification task (i.e., whether or not a USDA item is mentioned in a meal description), and 2) rank all the possible USDA hits to determine the top-5 matches. We then discuss our character-based models for handling typos.

The training condition (binary verification) differs from the test condition (ranking) because our data does not have information about which meal tokens map to which USDA foods (i.e., food segments are not labeled), so the model must learn this relation automatically through binary verification. We experimented with directly predicting the matching USDA foods for a given meal using a softmax output, but this resulted in much lower performance (3.6% top-5 recall on sandwiches, versus 76.5% with the ranking scheme we will describe later).

http://dx.doi.org/10.21437/Interspeech.2017-422

3.1. Convolutional Neural Network

As shown in Figure 1, our new model is composed of a shared 64-dimension embedding layer, followed by one convolution layer above the embedded meal description and max-pooling over the embedded USDA input (rather than one CNN for the meal and another for the USDA food item). The text is tokenized using spaCy (https://spacy.io). The meal CNN computes a 1D convolution of 64 filters spanning a window of three tokens with a rectified linear unit (ReLU) activation. During training, both the USDA input's max-pooling and the CNN's convolution over the meal description are followed by dropout [21] of probability 0.1[1] and batch normalization [22] to maintain a mean near zero and a standard deviation close to one. A dot product is performed between the max-pooled 64-dimension USDA vector and each 64-dimension CNN output of the meal description. Mean-pooling across these dot products yields a single scalar value, followed by a sigmoid layer for final prediction.[2] This design is motivated by our goal to compare the similarity of specific words in a meal description to each USDA food.

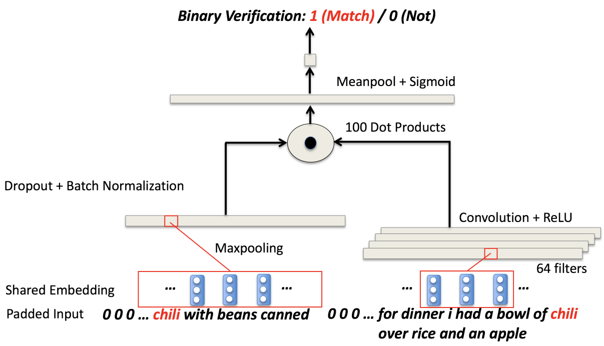

Figure 1: *Architecture of our CNN model for predicting whether a USDA food entry is mentioned in a meal description.*

To prepare the data for training, we padded each text input to 100 tokens[3] and limited the vocabulary to the most frequent 3,000 words, setting the rest to UNK. We trained the model to predict each (USDA food, meal) input pair as a match or not (1 or 0) with a threshold of 0.5 on the output. The model was optimized with Adam [23] on binary cross-entropy loss, norm clipping at 0.1, a learning rate of 0.001, early stopping after the loss stops decreasing for the second time on the validation data (i.e., 20% of the data), and mini-batches of 16 samples. We removed all capitalization and commas from the USDA foods.

3.2. Ranking USDA Foods

While the CNN model learns to predict whether or not a USDA food is mentioned in a meal, it does not directly retrieve the matching USDA foods from a meal. To make predictions at test time, we rank all the USDA food database entries, and we return the top-5 to the user. The most intuitive approach is to rank all USDA foods simply based on the posterior probability output by the model, but this does not accomplish our goal of providing the top-5 alternatives for each matching food in the meal, since the match is based on similarity with the entire meal description and does not allow us to distinguish which USDA hits match which eaten foods.[4]

Thus, we first perform semantic tagging on the tokens in the meal description with a pre-trained CNN tagger [24], which labels tokens as Begin-Food, Inside-Food, Quantity, and Other. Then, we feed the meal description into the pre-trained embedding layer in the model described in Section 3.1 to generate vectors for each token. Finally, we average the vectors for tokens in each tagged food segment (i.e., consecutive tokens labeled Begin-Food and Inside-Food), and compute the dot products between these food segments and each previously computed and stored USDA food vector.[5] The dot products are used to rank the USDA foods in two steps: a fast-pass ranking followed by fine-grained re-ranking that weights important words more heavily. For example, simple ranking would yield generic milk as the top hit for 2% milk, whereas re-ranking focuses on the property 2% and correctly identifies 2% milk as the top match.

- **ranking:** initial ranking of USDA foods using dot products between USDA vectors and food segment vectors.
- **re-ranking:** fine-grained word-by-word similarity ranking of the top-30 hits with a weighted distance D.

$$D = \sum_i \alpha_i \max_j (w_i \cdot w_j) + \frac{1}{N} \sum_j \beta_j \max_i (w_i \cdot w_j) \quad (1)$$

where N refers to the length of the tagged food segment. The left-hand term finds the most similar meal description token w_j to each USDA token w_i, weighted by the probability α_i that token was used to describe the USDA food item in the training data. In the same way, the right-hand term finds the most similar USDA token w_i to each meal token w_j, weighted by the probability β_j that token w_j was used to describe that USDA food item in the training data (see an example in Fig. 2).[6]

j = 0 chili	$\beta_0 + \alpha_0$	α_1(chili · with)	α_2(chili · beans)	α_3(chili · canned)
	i = 0 chili	i = 1 with	i = 2 beans	i = 3 canned

Figure 2: *A reranking example for the food "chili" and matching USDA item "chili with beans canned." There is only one β_0 term in the right-hand summation of equation 1, since there is only a single token "chili" from the meal description.*

3.3. Character-based Models

The aim of this work is to handle out-of-vocabulary (OOV) words, which the current word-based CNN models are unable to. For example, if the user misspells "cheerios" as "cherios," ideally the system would be able to correctly interpret this unknown word as the cereal "cheerios." To do this, we apply a character-based CNN model that learns word embeddings based on a convolution over characters, rather than mapping every unknown word to UNK, which often yields incorrect results.

[1]Performance was better with 0.1 dropout than 0.2 dropout or without any dropout.

[2]The inverse (mean-pooling before dot product) hurt performance.

[3]We selected 100 as an upper bound since the longest meal description in the data contained 93 words.

[4]The mean average precision is also lower (e.g., 12.7 on breakfast data versus 31.8 with the ranking method we will describe next).

[5]Our approach with CNN-learned embeddings significantly outperforms re-ranking with skipgram embeddings [25]. For comparison, on breakfast descriptions, our model achieves 64.8% top-5 recall, whereas re-ranking with skipgrams only yields 3.0% top-5 recall.

[6]Although the sum appears biased toward longer USDA foods, the right-hand term is over each token in the food segment, which is fixed, and the left-hand term is normalized by the α weights. Dividing D by the number of tokens in the USDA food item hurt performance (39.8% recall on breakfast data versus 64.8% with the best model).

3.3.1. *Levenshtein Edit Distance*

As a baseline, we implemented the Levenshtein edit distance [26] for determining the distance between two sequences of characters. The minimum distance is computed through the following recurrence relation, where each term corresponds to deletion, insertion, and substitution of a character, respectively:

$$d[i,j] = \min\left(d[i-1,j]+1, d[i,j-1]+1, d[i-1,j-1]+c\right) \tag{2}$$

where cost c is 0 if the two characters are equal and 1 if not. During the ranking step, if an OOV word is encountered, Levenshtein edit distance is used to determine which USDA foods are most similar to the tagged food segment containing the OOV.

3.3.2. *Character-based Embeddings*

The second method we explored for handling OOV words is a character-based CNN (charCNN) that processes each character in a token in order to generate a word embedding. As shown in Figure 3, each token is padded to 24 characters and fed through the shared network, which consists of a 15-dimension embedding layer over each character, followed by a convolution of 64 filters over windows of three characters with hyperbolic tangent activation,[7] max-pooling to reduce the 24 embeddings to a single embedding, and finally a highway network on top [27] to allow information flow along highways connecting layers.[8]

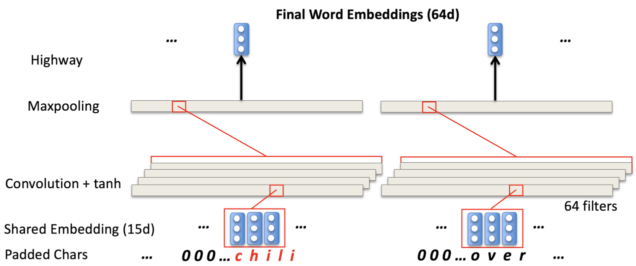

Figure 3: *Architecture of the character-based CNN model.*

Since the character-based models predict USDA matches primarily based on similar character sequences, rather than semantic similarity (e.g., “cheerios” would have a high similarity to “cheese”), we modified the USDA food database names during the ranking step for the DTW and charCNN models. Specifically, we shortened the USDA food names to more concise descriptions that only included those tokens that had an α weight above a threshold of 0.25. Thus, the new USDA names only included tokens that were used most often in the training data to describe those USDA foods, which are more likely to match the characters of that food in a user's meal description. For example, the USDA food “Cereals ready-to-eat, GENERAL MILLS, CHEERIOS” would become simply “cheerios,” which exactly matches the characters in the user-described food “cheerios.”

4. Experiments

To train and evaluate our models, we divided the data we collected on Amazon Mechanical Turk (AMT) into 80% training and 20% testing sets. For each positive USDA hit, we randomly selected a negative sample from the USDA items that were not mentioned in the meal. Every model we trained, except the charCNN, used additional training examples of meal descriptions without punctuation. Since the ranking method at test time matches USDA hits to individual food segments, we also added training examples containing only the tokens in the meal description that were aligned with the USDA food, as predicted by a Markov model (see our prior work [5] for details). For example, in addition to training on the full meal “I had a bowl of chili and an apple,” the models were also trained on the aligned segments “chili” and “apple.” With the augmented data, the performance is slightly higher, so we used alignments for all models except the charCNN. Unless we explicitly mention using the character-based models, we show results for the word-based CNN in all experiments.

We tested the models on both a binary verification task (i.e., how well the model predicts whether a meal matches a USDA food item) as well as ranking (i.e. how highly the model ranks the correct USDA matches for a given meal). The binary verification results are reported as accuracy, while for ranking we report top-5 recall and mean average precision (MAP) scores.

4.1. Data

As described in our prior work [5], we asked workers on AMT to generate meal descriptions that matched a selected subset of USDA items, which enabled us to build models that directly map from meal descriptions to USDA foods. To generate intuitive meal description tasks, we partitioned the over 5k foods in the USDA database into specific meals such as breakfast and dinner. We showed Turkers images of the foods to encourage more natural food descriptions. We also selected a set of 101 food items that we are using for a pilot user study. In total, the combined all-food category contains 31,712 meal descriptions spanning 5,124 USDA food items. In addition, we are currently collecting meal descriptions for a larger University of Minnesota Nutrition Coordination Center (NCC) Food and Nutrient Database [28] with over 18k foods.

4.2. Model Comparison

First, we show in Table 1 that the new CNN architecture with ranking outperforms the baseline CNN architecture [5] on the 101 foods case study. The only differences between the new model and the baseline are that the baseline feeds both the USDA and meal description inputs into a CNN (rather than only max-pooling the USDA input), the embeddings are dimension 50 instead of 64, the USDA input is limited to 20 tokens instead of 100, and a finite-state transducer (FST) is used instead of the ranking algorithm by automatically predicting which three USDA foods are aligned with the meal description. By ranking the USDA hits according to their dot product with each of the tagged food segments, rather than using the FST, we can boost the top-5 recall from 91.9 to 96.2. Using the new CNN, we boost the recall even further to 98.5.

We observe even clearer gains when we apply the new CNN architecture and re-ranking algorithm to the larger all-food dataset, which is what the system will be using at test time with real users. In Table 2, the top row applies the baseline CNN with ranking to predict the top-5 USDA matches. Its top-5 recall is approximately one-third that of the newer CNN in the other three rows. The last row, which ranks foods from both the USDA and Minnesota databases, still has comparably high performance despite the presence of 18k foods.

[7] We also tried rectified linear unit, but this did not perform as well.
[8] Without the highway layer, the performance is not as good.

Table 1: *FST decoding compared to the new ranking method described in this work on 101 foods data. These results are high because we collected 16k meals for 101 foods, whereas we only used 4k for the other meals with up to 2570 foods.*

Model	MAP	Recall
Baseline CNN + FST decode	86.9	91.9
Baseline CNN + ranking	83.4	96.2
New CNN + ranking	87.8	**98.5**

Table 2: *Evaluation of two CNN models (baseline and the newer model presented in this work) and two ranking approaches (ranking once vs. re-ranking) on the all-food dataset.*

Model	DB	MAP	Recall
Baseline CNN + ranking	USDA	6.73	20.7
New CNN + ranking	USDA	20.6	58.1
New CNN + re-ranking	USDA	**31.3**	**64.0**
New CNN + re-ranking	USDA + Minn.	31.2	63.8

4.3. Ranking Performance per Meal

We evaluated the best model (i.e., the new CNN trained on full meal descriptions plus aligned segments, with re-ranking for retrieving USDA hits) on all eight meals in addition to the all-food dataset, where each model is trained/tested only on its meal category. Note that there is a correlation between the number of foods in the meal category and the difficulty of the ranking task, since dinner has more foods than any other meal (2,570) and has the lowest MAP and recall scores, whereas salads have the fewest foods (232) and achieve a recall of 92.5.

Table 3: *Binary verification performance on training and held-out test data, MAP scores, and top-5 USDA recall per meal.*

Meal	Train Acc.	Test Acc.	MAP	Recall
Breakfast	93.1	84.4	31.8	64.8
Dinner	91.7	84.4	20.0	45.1
Salad	94.5	90.6	66.6	92.5
Sandwiches	91.2	86.3	37.5	76.5
Smoothies	93.7	87.5	39.3	75.1
Pasta/Rice	91.9	85.6	21.8	51.9
Snacks	92.8	85.6	32.3	67.3
Fast Foods	93.5	85.7	34.1	70.3
All Foods	96.0	94.1	31.3	64.0

4.4. Character-based Models

To evaluate how well the standard word-based model compares to the character-based models on noisy data, we artificially induced typos in the test set. The original test data was clean so the performance was the same with the standard word-based model as well as when we augmented it with character-based models. In Table 4, the last two rows apply the word-based model if there are no OOVs, use character-based models for food segments consisting of one OOV token, and combine the rankings generated by both models for food segments containing OOV and in-vocabulary words by taking the maximum similarity value per USDA food. We constructed a test set where 10% of the characters in each meal description were randomly deleted, substituted, or inserted. The error type, the index of the modification, and the new character for substitutions and insertions were all randomly selected. We see that the charCNN has the highest MAP and top-5 recall scores on the noisy test data.

Table 4: *Comparison of character-based approaches to the basic word-based CNN model on the all-food data with 10% typos.*

Model	MAP	Recall
Baseline Levenshtein	6.27	15.5
Char CNN	8.65	17.1
Word CNN	12.0	26.3
Word CNN + Levenshtein	13.4	29.1
Word CNN + Char CNN	**13.8**	**30.0**

5. Discussion

To better understand the behavior of our models, we qualitatively analyze the top predicted USDA hits at test time, as well as the learned embeddings. In Table 5, we observe reasonable top-3 USDA foods predicted for each tagged food segment by the word-based model augmented with charCNN for the meal "I had a bowl of cherios with a banana and a glass of 1% milk." Note that since "cheerios" is misspelled as "cherios," the word model would predict matches for UNK as it is an OOV word; however, the charCNN correctly handles the error. In addition, we demonstrate that the charCNN is able to learn meaningful word embeddings. In Table 6, we see that the top-5 neighbors of the USDA food item are intuitive, where we computed the nearest neighbors by minimizing the Euclidean distance between a given food's embedding and all the other USDA foods.

Table 5: *Top-3 predicted USDA hits for tagged foods in the meal "I had a bowl of **cherios** with a **banana** and a glass of **1% milk**."*

Food	Top-1	Top-2	Top-3
cherios	cheerios	oat cluster cheerios	frosted cheerios
banana	banana	banana pudding	banana pepper
1% milk	1% milk	dry whole milk	milk low sodium

Table 6: *Top-5 nearest USDA neighbors, based on Euclidean distance, to the food "beef new zealand imported tongue raw."*

Top-5 USDA Foods
beef new zealand imported kidney raw
beef new zealand imported inside raw
beef new zealand imported sweetbread raw
beef new zealand imported heart raw
beef new zealand imported manufacturing beef raw

6. Conclusion

We have demonstrated that our new CNN architecture with ranking outperforms the baseline CNN model with FST decoding. In addition, we have shown that character-based models effectively handle misspelled words. In our future work, we will explore jointly learning word and character embeddings, and we will evaluate our models on spoken data more similar to how the deployed system will be used by real users at test time.

7. References

[1] Y. Wang and M. Beydoun, "The obesity epidemic in the United States–gender, age, socioeconomic, racial/ethnic, and geographic characteristics: A systematic review and meta-regression analysis," *Epidemiologic reviews*, vol. 29, no. 1, pp. 6–28, 2007.

[2] C. Ogden, M. Carroll, B. Kit, and K. Flegal, "Prevalence of childhood and adult obesity in the United States, 2011-2012," *Jama*, vol. 311, no. 8, pp. 806–814, 2014.

[3] E. Finkelstein, J. Trogdon, J. Cohen, and W. Dietz, "Annual medical spending attributable to obesity: Payer-and service-specific estimates," *Health affairs*, vol. 28, no. 5, pp. w822–w831, 2009.

[4] J. Ingber, "My fitness pal: A guide to an accessible fitness tool," 2014.

[5] M. Korpusik, Z. Collins, and J. Glass, "Semantic mapping of natural language input to database entries via convolutional neural networks," *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[6] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass, "Data collection and language understanding of food descriptions," *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014.

[7] M. Korpusik, C. Huang, M. Price, and J. Glass, "Distributional semantics for understanding spoken meal descriptions," *Proceedings of 2016 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint arXiv:1606.01781*, 2016.

[10] Y. Xiao and K. Cho, "Efficient character-level document classification by combining convolution and recurrent layers," *arXiv preprint arXiv:1602.00367*, 2016.

[11] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," 2016.

[12] Z. Wang, H. Mi, and A. Ittycheriah, "Sentence similarity learning by lexical decomposition and composition," *arXiv preprint arXiv:1602.07019*, 2016.

[13] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proceedings of Advances in neural information processing systems (NIPS)*, 2014, pp. 2042–2050.

[14] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *arXiv preprint arXiv:1512.05193*, 2015.

[15] W. Yin, S. Ebert, and H. Schütze, "Attention-based convolutional neural network for machine comprehension," *arXiv preprint arXiv:1602.04341*, 2016.

[16] W. Ling, I. Trancoso, C. Dyer, and A. Black, "Character-based neural machine translation," *arXiv preprint arXiv:1511.04586*, 2015.

[17] Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[18] C. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014, pp. 1818–1826.

[19] Y. Kim, Y. Jernite, D. Sontag, and A. Rush, "Character-aware neural language models," *arXiv preprint arXiv:1508.06615*, 2015.

[20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of Advances in neural information processing systems (NIPS)*, 2015, pp. 649–657.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] M. Korpusik and J. Glass, "Spoken language understanding for a nutrition dialogue system," *IEEE Transactions on Audio, Speech, and Language Processing*, 2017.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119.

[26] A. Marzal and E. Vidal, "Computation of normalized edit distance and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 926–932, 1993.

[27] R. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[28] I. M. Buzzard and D. Feskanich, "Maintaining a food composition data nase for multiple research studies: The NCC food table," *Rand et al., reference*, vol. 72, pp. 115–122, 1987.