Spoken Language Understanding for a Nutrition Dialogue System

Mandy Korpusik, *Student Member, IEEE*, and James Glass, *Fellow, IEEE*

***Abstract*—Food logging is recommended by dieticians for prevention and treatment of obesity, but currently available mobile applications for diet tracking are often too difficult and time-consuming for patients to use regularly. For this reason, we propose a novel approach to food journaling that uses speech and language understanding technology in order to enable efficient self-assessment of energy and nutrient consumption. This paper presents ongoing language understanding experiments conducted as part of a larger effort to create a nutrition dialogue system that automatically extracts food concepts from a user's spoken meal description. We first summarize the data collection and annotation of food descriptions performed via Amazon Mechanical Turk (AMT), for both a written corpus and spoken data from an in-domain speech recognizer. We show that the addition of word vector features improves conditional random field (CRF) performance for semantic tagging of food concepts, achieving an average F1 test score of 92.4 on written data; we also demonstrate that a convolutional neural network (CNN) with no hand-crafted features outperforms the best CRF on spoken data, achieving an F1 test score of 91.3. We illustrate two methods for associating foods with properties: segmenting meal descriptions with a CRF, and a complementary method that directly predicts associations with a feed-forward neural network. Finally, we conduct an end-to-end system evaluation through an AMT user study with worker ratings of 83% semantic tagging accuracy.**

***Index Terms*—Conditional random field, crowdsourcing, neural networks, semantic tagging, word vectors.**

I. Introduction

EXCESSIVE weight is becoming a serious health concern. It leads to diseases such as obesity and diabetes, which can cause complications requiring expensive medical treatment. In the United States, one-third of adults over 20 are obese and nearly 70% overweight, leading to annual healthcare costs of $113.9 billion [1], [2]. Food journaling is an effective way to combat weight gain, but existing diet tracking applications are often too tedious and cumbersome for patients to use [3].

The solution we propose is an artificially intelligent diet tracking application powered by speech and language understanding technology that makes self-assessment of nutrient and caloric intake quick and easy (see Fig. 1). Users simply describe their meal in natural language via speech or text, and the system automatically determines the nutrition facts.

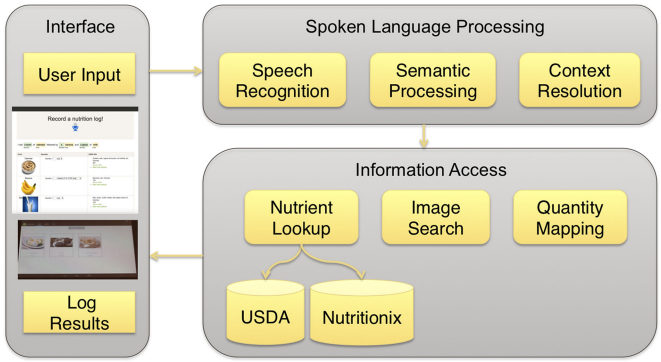

Fig. 1. A diagram of the nutrition system's flow, illustrating the process of the user recording a meal, followed by spoken language understanding, nutrient database lookup, and finally responding to the user with the results.

In this paper, we investigate the core research questions underlying development of our system's natural language processing (NLP) technology. Specifically, we address the challenge of automatically identifying the food items in a user's spoken meal description (e.g., extracting the food "a bowl of Kellogg's cereal" from the meal "This morning for breakfast I had a bowl of Kellogg's cereal"), in order to map the foods to a database of nutrition facts. If we view a food item as an "entity," or foods and properties (e.g., brands and quantities) as "slots," we can map this task to a well-known named entity recognition (NER) or slot filling problem. Thus, we investigate NER and slot filling methods for our task.

Other research questions we investigate involve how to train and test our models. That is, what data do we need, and how do we collect them? Can we handle spoken meals in addition to written meals? Does borrowing ideas from the field of distributional semantics help? Can we identify foods without manual feature engineering? How do we determine which properties go with which food items, and what if there is a long-distance dependency between a food and its property? Finally, how do we evaluate our system's performance?

Our exploration of these research challenges yields the following findings:

1) State-of-the-art NER and slot filling methods can be successfully applied to a new domain for food entity extraction on both written and spoken meals.

Manuscript received August 1, 2016; revised February 8, 2017; accepted April 11, 2017. Date of publication April 17, 2017; date of current version June 5, 2017. This work was supported in part by a grant from Quanta Computing, Inc., in part by the NIH, and in part by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ani Nenkova. *(Corresponding author: Mandy Korpusik.)*

The authors are with the MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139 USA (e-mail: korpusik@mit.edu; glass@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2694699

2) Incorporating word vector features, inspired by distributional semantics, improves performance with a conditional random field (CRF) model.
3) A neural network that does not require any feature engineering or pre-trained NLP tools outperforms a feature-based CRF classifier on semantic tagging of spoken data.
4) Data for a new domain can be collected quickly and reliably on Amazon Mechanical Turk (AMT); similarly, a deployed system's performance can be quickly evaluated on AMT without conducting lengthy user studies.
5) Food-property association is more accurately determined with a classifier (e.g., random forest or feed-forward neural network) that incorporates long-distance dependency features, rather than locally segmenting a meal.

In the remainder of this paper, we begin by discussing the initial prototype of the nutrition system and related work. Section IV details data collection for a written corpus, as well as a spoken corpus from an in-domain speech recognizer, Section V discusses semantic tagging and food-property association, Section VI presents experimental results and analysis, and Section VII concludes.

II. The Nutrition System

The understanding component forms part of a larger nutrition logging prototype [4], [5] whose current interface displays the output of a speech recognizer given the user's spoken input utterance, along with color-coded semantic tags (e.g., quantity, brand, etc.) associated with particular word sequences. The segmented food concepts are then shown in a table along with potential matches to a nutritional database containing over 20 000 foods from the USDA and other sources.

The flow of the nutrition system is shown in Fig. 1. After the user generates a meal description by typing or speaking, the language understanding component labels each token in the description and assigns properties (i.e., "brand," "quantity," and "description") to the corresponding "food" tokens. We used conditional random field (CRF) and neural network (NN) models for the language understanding tasks: semantic tagging and food-property association. The language understanding output is used for database lookup and image search before responding to the user.

III. Related Work

To motivate our work on semantic tagging in the food domain, we start by introducing prior work on the similar tasks of spoken language understanding (SLU) in dialogue systems (i.e., determining user intent and slot filling, or tagging specific words in a user's query as values for slots such as arrival or departure city in the air travel domain). We also examine the closely related task of named entity recognition and classification (NERC). Finally, we present work in spoken dialogue systems (SDS) and distributional semantics.

A. Spoken Language Understanding

The SLU literature largely focuses on the Air Travel Information Systems (ATIS) corpus [6], [7], which is composed of spoken queries about flight information. For example, understanding the query "I want to fly to Boston from New York next week" involves identifying the goal as airfare and slot values of Boston, New York, next, and week for the slots departure city, arrival city, relative departure time, and departure time period respectively.

Research on ATIS has moved from early work involving handmade template matching requiring expensive grammars to more data-driven methods. He and Young [8] showed improved performance using the expectation maximization algorithm for a generative hidden vector state (HSV) model over hand-crafted semantic grammar rules. Wang *et al.* [9] used a discriminative CRF rather than generative models to reduce slot error rate by over 20%. Raymond *et al.* [10] similarly demonstrated that the addition of a-priori long-term dependency features in CRF models led to better performance than the generative finite-state transducer, and Meza-Ruiz *et al.* [11] also showed that global dependency features in discriminative models outperform the generative HSV model. This motivates our use of the CRF with its sequential processing. Heintze *et al.* [12] demonstrated a performance improvement as incrementally longer utterance prefixes are seen by a classifier, and Tur *et al.* [13] used dependency parsing to simplify natural utterances into more concise, keyword-style queries that are easier for classifiers to process. Tur *et al.*'s 2010 study [14] analyzing the state-of-the-art on ATIS revealed common error patterns that were still unresolved, including long-distance dependencies in slot filling.

Recently, neural networks such as bidirectional RNNs [15] and LSTMs [16] have been shown to outperform CRFs, which motivates our use of neural networks on our tasks.

A similar trend has been observed for NERC as for SLU, where early work applied hand-crafted rules [17], but transitioned to machine learning methods over the course of the 1990s and early 2000s, including supervised learning (e.g., hidden Markov models, support vector machines, and CRFs [18], [19]), as well as unsupervised clustering and semi-supervised learning from distributional semantics, given a set of seed entities (which motivates our prototype features for semantic tagging). Again, more recently, neural networks have been investigated, including convolutional neural networks (CNNs) [20] and LSTM-CRFs [21], [22].

B. Spoken Dialogue Systems

Since our ultimate goal is to build a spoken dialogue system (SDS) that responds to a user and asks followup clarification questions after they record a meal description, we also discuss state-of-the-art methods for dialogue systems. These typically involve either statistical models or neural network approaches. Whereas previously dialogue managers were often rule-based [23], requiring expensive hand labeling by experts and increasing complexity as more actions were incorporated into the set of dialogue states, recent implementations of dialogue managers are typically statistical in nature [24], [25]. One method is to use a Partially Observable Markov Decision Process (POMDP) [26], in which reinforcement learning

selects optimal policies through rewards based on the state of the environment.

Newer neural methods for conversational agents apply a sequence-to-sequence learning method commonly used in machine translation [27] to short-text conversation (i.e., single turns in a dialogue) [28], [29]. These researchers focus on different aspects of conversational agents, such as incorporating context from previous turns in the conversation [30], enhancing the diversity of the system's generated responses [31], or endowing the system with a personality so that its responses are more consistent [32].

C. Distributional Semantics

Vector space models of semantics [33] attempt to teach computers meanings of words or documents through their relationship in vector space. For example, words with a similar meaning can be represented by vectors that lie close to each other in vector space. Recent work [34]–[36] has shown that using word vectors as features in classifiers can improve natural language processing performance in a variety of tasks, such as part-of-speech tagging in multiple languages [37], enriching spoken queries in dialogue systems [38], and semantic tagging [39]. Motivated by this finding, we investigated two approaches for incorporating word vector features into a CRF semantic tagging model: using dense vector values directly, and measuring the cosine distance between tokens and "prototypes" (i.e., words most representative of a category).

Finally, we compare our work to a similar approach taken by Manurinakurike *et al.* [40] for a different task involving understanding language referring to visual objects in scenes. Specifically, one speaker describes the objects in a scene, and another person identifies which scene among several options is the correct match. Their language understanding pipeline parallels ours, since they first segment scene descriptions with a CRF (like our semantic tagging with a CRF), followed by association of visual objects and word segments using a logistic regression classifier (like our food-property association with a random forest classifier). However, they use visual features, whereas we rely on linguistic features.

As we have seen in this section, related work in SLU, NERC, and SDS applies CRFs and neural networks, which motivates our work on a similar task, but in a completely different domain. Our results follow a similar conclusion, that neural networks perform comparably well (or even better than) CRFs, without requiring any manual feature engineering.

IV. Data Collection and Annotation

In order to train our language understanding models, we needed to collect meal descriptions, both written and spoken, where each token was labeled as a property (e.g., brand or quantity), and property tokens were assigned to food tokens (e.g., the quantity "bowl" was assigned to the food "cereal"). This section illustrates the process of using an online crowdsourcing platform to collect a new corpus in the nutrition domain, as well as techniques for reducing noise in the data.

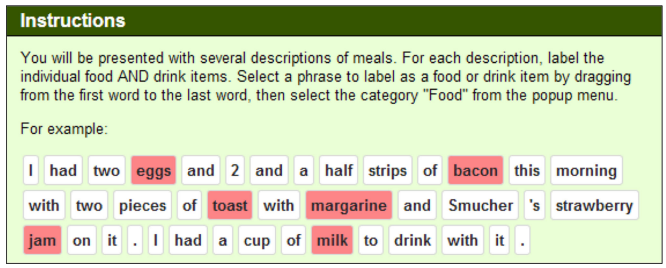

Fig. 2. AMT task instructing workers to label each food word.

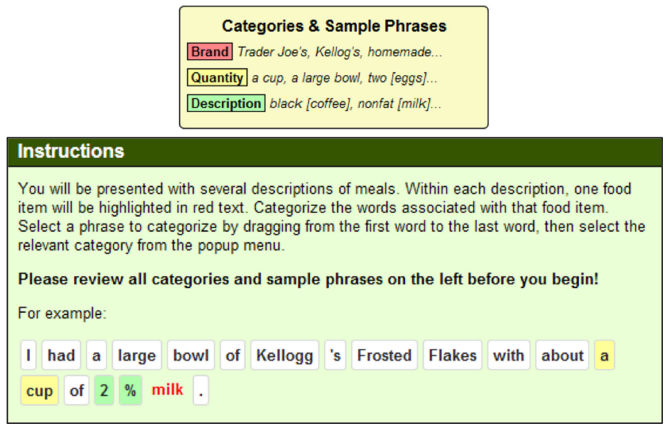

Fig. 3. After Turkers labeled the foods, the final AMT task asked them to label properties (e.g., quantities and brands) of a given food word.

A. Text Corpus

We deployed three subtasks of experiments on Amazon Mechanical Turk (AMT) in order to crowdsource our data collection and annotation, since this is a source of demographically diverse participants, and high-quality data can be obtained inexpensively and rapidly [41]. In the first phase, we prompted Turkers to write a description of a meal (breakfast, lunch, dinner, or snack) as they would describe it orally:

> *Please record what you ate for breakfast today (or yesterday) using as much detail and accuracy as possible. Be creative - we will not accept repeat answers. Try to include as much additional information as you remember, such as brand names, preparation methods, portion sizes, etc. Please write the description in the way you'd imagine describing it orally. Example: I had a boiled egg, a Thomas's English muffin, and an ounce of organic butter. I had a cup of coffee French roast doctored with an ounce or two of half-and-half and two teaspoons of brown sugar.*

The diaries were then tokenized and used as input for the second phase, where we asked Turkers to label individual food items within the diaries (Fig. 2). The third phase combined the meal descriptions with their food labels and prompted Turkers to label the brand/quantity/description properties associated with a particular food item (Fig. 3) [42].

We collected and labeled a total of 22,000 meal descriptions including breakfast, lunch, dinner, and snacks on AMT, which we used to train our models. The frequency of each tag is shown in Table I. We measured the reliability of the data annotations by calculating the inter-annotator agreement among Turkers. Specifically, we calculated Fleiss' kappa scores for the

TABLE I
STATISTICS FOR TOKENS ASSIGNED ONE OF FIVE POSSIBLE LABELS OUT OF THE 22,000 COLLECTED MEAL DESCRIPTIONS

Label	Frequency
Food	76,399
Brand	13,826
Quantity	38,668
Description	46,898
Other	89,729

two labeling tasks: 0.77 for food labeling, and 0.41 for property labeling. The score for the food labeling task indicates substantial agreement; as expected, the score for property labeling is lower, but the score still indicates a moderate amount of agreement [43].

We also incorporated algorithms for improving Turker labeling performance. In order to determine whether the food and property labels selected by the Turkers were reasonable, we automatically detected which tokens were foods or properties in each meal description and required Turkers to label these tokens upon submitting a property labeling task. If a token was missing, the submission error message would require the Turker to return to the task to complete the labeling more accurately, but would not reveal which tokens were missing.

To automatically generate hidden food and property labels, we used a trie matching algorithm [44] trained on the USDA food lexicon. A trie is an n-ary tree data structure where each node is a character, and a path from the root to a leaf represents a token. We built a variant of the standard trie where each node contains a token that is part of a USDA food entry, and a path from the root to a leaf represents an entire food phrase. For example, a node might contain the token "orange," and its child node might contain the token "juice." Then, the matching algorithm would find every matching entry from the USDA trie that is present in a meal description. Since USDA food entries often contain only the singular form of a food token, we incorporated plural handling into the trie matching, using the Evo Inflector libary's implementation of Conway's English pluralization algorithm [45].

B. Speech Corpus and Recognizer

The experiments presented in prior work [42] relied upon written, rather than spoken, data, whereas at test time the system must be able to handle spoken user input. To address this limitation, we collected a corpus of spoken meal descriptions, and created a nutrition speech recognizer [46]. We collected the speech data via AMT [47], where we asked Turkers to record 10 meal descriptions. The diaries were selected from previously collected written meal descriptions, and spelling and grammar errors were manually corrected. The Turkers' recording, converted to text via a recognizer embedded in the AMT task, was required to contain at least 60% of the words in the transcript they were reading in order to submit the task.

We split the resulting 2,962 utterances (from 37 speakers totaling 2.74 hours) into 80% training, 10% development, and 10% test sets, and removed punctuation and capitalization from the text data for training the language model. Using Kaldi [48], we trained a fully-connected, feedforward, deep neural network (DNN) acoustic model and a trigram language model on 40,000 written sentences (this is a larger set than the original 10,000 meal logs [46] because each meal was split into individual sentences). The DNN's input features were Mel-frequency cepstral coefficients (MFCCs) that are the standard for speech recognition. The network was used in conjunction with a hidden Markov model (HMM) recognizer that had 265 tied states; therefore it had 265 outputs. The DNN had 6 hidden layers, each with a sigmoid nonlinearity, followed by a softmax. The decoder had a word error rate (WER) of 7.98% on the test set. We then annotated the semantic tags and food-property associations of the recognizer's output on AMT, as described in [4] for subsequent understanding evaluation.

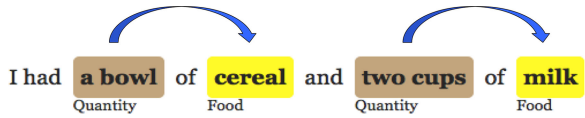

Fig. 4. A depiction of the two language understanding tasks: semantic tagging (e.g., quantities and foods), followed by food-property association (e.g., assigning "a bowl" to "cereal").

V. LANGUAGE UNDERSTANDING

In the nutrition system, after the user describes his or her meal, the language understanding component must not only identify the foods and properties (i.e., semantic tagging), but also determine which foods are associated with which properties (e.g., selecting "milk" as the food which "two cups" describes, rather than the preceding food "cereal" in Fig. 4). This section presents the primary research challenges we encountered in order to solve the two language understanding tasks: semantic tagging and food-property association.

A. Semantic Tagging

To address the first language understanding task in the system (i.e., labeling of each token in a meal description as a food, quantity, brand, or description), we viewed it as a type of named entity recognition. Thus, we followed the approaches used on this task and first applied a standard conditional random field (CRF) baseline model, to which we investigated the addition of more complex feature sets involving word vectors. Finally, we examined whether we could solve the semantic tagging problem without any feature engineering by using a convolutional neural network (CNN).

1) Tagging With CRFs: CRFs are useful models for natural language processing tasks, such as slot filling, that involve sequential classification [49]. In such a problem, we wish to predict a vector of output labels $\vec{y} = \{y_0, y_1, ..., y_T\}$ corresponding to a set of input feature vectors $\vec{x} = \{\vec{x_0}, \vec{x_1}, ..., \vec{x_T}\}$. Due to complex dependencies, this multivariate prediction problem is challenging. We can use graphical models to represent a complex distribution over many variables more easily. The structure of the graphical model determines how the probability

distribution factorizes, based on a set of conditional independence assumptions [49].

In the past, generative models, such as the naive Bayes classifier and hidden Markov models (HMMs), were popular. They describe how to "generate" values for features given the label. However, since they model a joint probability distribution $p(\vec{y}, \vec{x})$, these models can become intractable when there are complex dependencies. The CRF takes a discriminative approach, where the conditional distribution $p(\vec{y}|\vec{x})$ is modeled directly. The linear-chain CRF has the form

$$Pr(\vec{y}|\vec{x}, \theta) = \frac{1}{Z(\vec{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \vec{x_t}) \right\} \quad (1)$$

where θ_k is a weight parameter for feature function f_k, and $Z(\vec{x})$ is the normalization factor.

The baseline features included n-grams, part-of-speech (POS) tags (e.g., whether the token is a noun, which is more likely for foods, or an adjective, which may correspond to brands and descriptions), and presence in a food or brand lexicon (i.e., whether the token under consideration appeared in either the list of USDA foods or brands). We can improve upon this set of features with semantic word vector features.

According to distributional semantics theory [50], [51], words with similar meanings appear in similar contexts and have similar vector representations, so we explored using neural network-trained vectors as CRF tagging features to account for semantics. A popular method for learning word embeddings is Mikolov's Skip-gram model [52], released as the word2vec toolkit,[1] which learns word vector representations that best predict the context surrounding a word. In our experiments, we trained the vectors with the continuous bag-of-words (CBOW) approach, which predicts the current word based on the context [53].

The research challenge here is to determine the best way of incorporating word vectors into CRF feature sets. First, we directly used vector component values as features for each of the 300 dimensions of the pre-trained word vectors from the Google News corpus, which has a three million word vocabulary from about 100 billion words total (available on the word2vec website). For these experiments, we used the CRFsuite [54] implementation rather than CRF++ (although performance was similar) for two reasons: faster running time and the ability to use vector float values.

Motivated by the linguistics literature in which Lakoff [55] argues that people categorize objects according to central members of a class (i.e., "prototypes"), and that other members of the same class have a degree of belonging which corresponds to their "similarity" to the prototype members, we also developed a prototype similarity feature. Thus, in addition to using the continuous, dense embeddings as features in our models, we explored a distributional prototype method [56] for discretizing the embedding features: representing each label category with a prototype word (e.g., "milk" for food) and using the similarity between a token and prototypes as features [56]. We experimented both with features representing the similarity between a token and individual prototypes, as well as the average similarity between a token and all the prototypes in a category. In addition, we explored binary features for similarities below a threshold δ tuned with cross-validation. The similarity was calculated with cosine distance, and the prototypes were selected through normalized pointwise mutual information (NPMI)

$$\lambda_n(\text{label},\ \text{word}) = \frac{\lambda(\text{label},\ \text{word})}{-\ln p(\text{label, word})}, \quad (2)$$

where $\lambda(\text{label, word})$ is the standard PMI

$$\lambda(\text{label},\ \text{word}) = \ln \frac{p(\text{label},\ \text{word})}{p(\text{label})p(\text{word})}. \quad (3)$$

For each label, the NPMI was computed for every vocabulary word. The top m words were chosen as prototypes for each label, where $m = 50$ was selected via cross-validation.

2) Tagging With Neural Networks: Although CRFs are a powerful discriminative classifier for sequential tagging problems, they require manual feature engineering. A new alternative which does not require any feature engineering is a neural network. In particular, recurrent neural networks and their long short-term memory (LSTM) variant that addresses the vanishing/exploding gradients problem [57], [58], have become popular in speech recognition and natural language processing, including semantic tagging [59], [60].

In addition, convolutional neural networks (CNNs), originally developed for computer vision, have recently been successfully applied to NLP tasks such as sentence matching [61] and machine comprehension [62]. Recent work has shown significant performance improvement over previous state-of-the art text classification techniques using very deep character-level CNNs [63]. Whereas for images the CNN learns filter maps that apply 2D convolutions over regions of images, CNNs can also learn filters that apply 1D convolutions to sequences of words in a sentence. A CNN window of 5 tokens can be interpreted as an n-gram of 5 tokens, which directly provides context similar to the features used in a CRF.

In our work, for the semantic tagging task we implemented variants of the CNN model in Keras [64]. Each model was composed of a word embedding layer initialized uniformly with 150 dimensions, followed by a number of CNN layers, and finally a fully-connected layer with a softmax activation to predict the semantic tag. We used the Adam optimizer [65], binary cross-entropy loss, and dropout with early stopping to prevent overfitting. We chose to focus on the CNN rather than the LSTM due to faster training and fewer parameters required. Lei *et al.*'s work on recurrent CNNs [66] demonstrates that models combining recurrence and convolution may perform even better on language tasks than either individually.

B. Food-Property Association

After semantic tagging, the second language understanding task is to determine which foods map to each property, which we call food-property association (see Fig. 4). The primary research question we address is whether this task can be accomplished more accurately by segmenting the meal description into food

[1] https://code.google.com/p/word2vec/

chunks (each containing a food item and its associated properties), or by directly predicting the most likely food for each property (i.e., non-segmental).

1) Segmental Approaches: The first four methods we investigated for associating foods with properties all involved segmenting meal descriptions into food chunks, where all properties within a segment were assigned to the food item that appears within the same chunk. For example, the meal description "I ate two pancakes and drank a glass of milk" would be segmented into two chunks: "two pancakes" and "a glass of milk." Our previous work [42] explored a simple rule that assigned properties to the subsequent food, a Markov model [67]–[69], transformation-based learning (TBL) [70], and a CRF that performed best.

In order to adapt TBL and CRFs to the property association problem, we framed it as a classification task. To do this, we modeled it after the noun phrase (NP) chunking problem, a well-known NLP task. In NP chunking, each word in a sentence belongs to one of three IOB classes: B (begins an NP), I (inside an NP), or O (outside an NP). For the food chunking problem, we used the same three classes to label each word as belonging to a food chunk or not. The features used by the classifier were composed of a token and its semantic tag (i.e., food, quantity, brand, description, or other).

2) Non-segmental Approaches: One drawback to using the segmental representation is that it assumes properties appear either directly before or after the food with which they are associated, neglecting long-range dependencies. For example, in the meal description "I had two eggs and cheese from Safeway," the brand "Safeway" should be assigned to both "eggs" and "cheese;" however, with the segmenting scheme, it is impossible to associate "Safeway" with "eggs" without also assigning the quantity "two" to "cheese" (since all properties are applied to all foods within a segment, and in this case there are either two separate segments for "eggs" and "cheese" or one segment for both). In addition, converting the labeled AMT data to IOB format requires making assumptions where some information (e.g., long-range dependencies) is omitted. Thus, we investigated whether an alternative method for food-property association that uses a classifier to predict which food a property describes might perform better.

In our approach, given a tagged meal description, for each of the property tokens the classifier determines with which food it is associated. Given a property token t_i, we iterate through each food token f_j in the meal description and generate features for each (t_i, f_j) pair. For each pair, the classifier outputs a probability that f_j is the corresponding food item for t_i. Then, for each t_i, the f_j with maximal probability is selected. Note that this does not allow a property to be associated with more than one food, but we consider this a first step and in future work will explore association of multiple foods via a vector of probabilities rather than one hard label.

The classifiers were trained using five features: the property token, whether the food token is before or after the property token, the distance between the two tokens, the property's semantic tag, and the dependency relation between the property and food tokens if the food is the property's head in the meal log's dependency parse tree. We explored three different classifiers, using the Scikit-learn toolkit's implementation for Python [71]: a random forest (i.e., a collection of decision tree classifiers trained on a random sample of training data), logistic regression, and a naive Bayes classifier. We used the spaCy NLP toolkit[2] in Python for dependency parsing, tokenizing, and tagging because it is fast and provides shape features (e.g., capitalization, numbers, etc.) that improved performance over our manually defined shape features. We found that the random forest classifier outperformed the rest, so we only show experimental results for the random forest. Performance was evaluated via F1 scores for property tokens.

Again, we asked whether it is possible to accomplish this task with a neural network. Therefore, similar to the CNN we applied to semantic tagging, we also built a feed-forward neural network for food-property association. The network is composed of one fully-connected layer with 128 hidden states and a sigmoid activation function, followed by the final output layer predicting whether or not the food and property pair is a match. It uses the same features as the random forest classifier, but instead of the string of the property token, it uses the embedding representation learned by the CNN for semantic tagging. Related work also used neural networks with positional features for relation detection [72].

VI. Experimental Results and Analysis

This section presents our results on the two language understanding tasks we have introduced and provides an analysis of the findings that our experiments reveal. For each task, we use our results to highlight the insights we obtained as answers to the research questions we asked in the introduction.

To evaluate our methods for labeling and associating foods and properties, we split the AMT data into training and test sets and computed the precision (i.e., the fraction of predicted labels that were correct), recall (i.e., the fraction of gold standard labels that were predicted), and F1 (i.e., the harmonic mean of precision and recall) scores for each approach. We then measured statistical significance in performance differences among several approaches using McNemar's significance test [73]. We present results on both text and speech data. We also show performance of the end-to-end system evaluated on AMT.

A. Semantic Tagging Experiments

For the semantic tagging task, we first discuss the initial experiments with CRFs on baseline features, followed by improvements gained by adding word vector features inspired by distributional semantics; we also investigate the effect of increasing the training data size. We demonstrate that a deep CNN outperforms the best CRF on spoken data, which is what the live system must handle at test time, without requiring any feature engineering or additional NLP tools.

Since the data obtained on AMT are noisy (see Table II), for evaluation we manually cleaned the test sets for the written and

[2]https://honnibal.github.io/spaCy/

TABLE II
EXAMPLES OF TURKER LABELING MISTAKES

Token	AMT Label	Expert Label
Whole Foods	Other	Brand
some	Other	Quantity
tangerine	Food	Description
a	Other	Quantity

TABLE III
CRF AND NN F1 SCORES PER LABEL ON THE SEMANTIC TAGGING TASK ON TEXT

Model	Food	Brand	Num	Descr	Avg
Baseline	85.4	76.7	92.8	77.6	85.4
Baseline CRF	94.2	84.1	94.9	90.0	91.9
Best CRF	**94.6**	**85.7**	95.1	**90.3**	**92.4**
1 CNN ($w = 5$)	93.8	82.9	95.0	89.1	91.4
2 CNNs ($w = 5, 5$)	94.3	83.2	94.9	89.8	91.7
3 CNNs ($w = 5, 5, 3$)	94.5	83.8	**95.5**	89.9	92.0

w refers to the CNN filter width per stacked CNN layer. The baseline predicts tags using the most frequent tag in the training data for a token (None if unseen in training).

spoken semantic tagging data. We noted a 1% performance improvement when evaluating on the expert-labeled test set rather than using the noisy AMT labels.

We initially applied the standard CRF model (the Python CRFsuite implementation [54]) to the semantic tagging task. This required extensive feature engineering, in which we began with a set of baseline features selected via cross-validation: the combination of n-grams, food and brand lexicon features [74], part-of-speech (POS) tags [75], and a shape feature indicating whether the token was in titlecase, lowercase, uppercase, a number, or a piece of punctuation. We subsequently added two new features based on learned word embeddings trained on large corpora. We explored training vectors on the Google News and Wikipedia corpora, as well as on domain-specific nutrition data (i.e., the compilation of meal diaries collected from AMT); however, the vectors trained on Google News performed best because the data size is much larger than the nutrition data set (1 billion versus 265,000 words).

We found that incorporating new word vector features improved performance over the baseline, confirming our hypothesis that features based on the distributional semantics theory would help. Specifically, we added two word vector features: the 300-dimensional word embeddings themselves and individual distributional prototype similarities since we found that using a unique feature for each prototype's similarity performed better than averaging all the similarities per category. The performance of the best CRF with all these features is shown in the third row of Table III.

In answer to our research question posed in the introduction about the ability of neural networks to perform well, we discovered that not only are CNNs competitive with CRFs on written data (Table III), but they even outperform the best CRF on spoken data (Table IV). The best CNN, with three layers of 64 filters each, scores only 0.4 points below the best CRF with

TABLE IV
SEMANTIC TAGGING PERFORMANCE ON SPOKEN DATA

Model	Food	Brand	Num	Descr	None	Avg
Best CRF	93.3	**79.0**	96.6	87.7	97.1	90.8
Best CNN	**93.9**	77.9	**97.5**	**89.1**	**98.1**	**91.3**

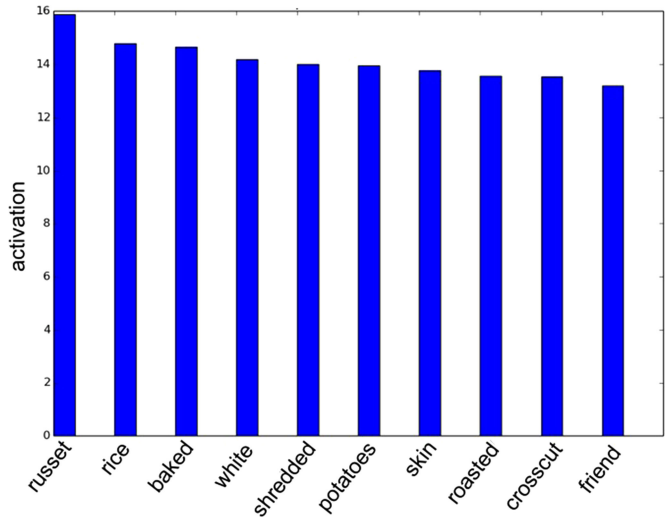

Fig. 5. The 10 words with the highest response to filter 52 of the CNN in the single layer model, which mostly relate to potatoes.

word vector features and outperforms the strong CRF baseline. This is encouraging since the CRF requires many handcrafted features, part-of-speech tags, and pre-trained word vectors, whereas the CNN is end-to-end in that it automatically learns its word embeddings and does not require additional NLP tools such as part-of-speech tagging. We also observe increased performance as we stacked more CNN layers, improving the average F1 score from 91.4 up to 92.0 with three layers.

On spoken data, the best CNN actually outperforms the best CRF (Table IV), demonstrating its ability to perform well despite speech recognition errors. The best CNN appears to be even more sensitive to misrecognized brands, causing lower performance on brand recognition, but is stronger overall. The poor performance on semantic tagging of brands for both corpora is likely due to the small number of brand tokens (i.e., only 3.4% of the test data's tokens are brands), as well as the difficulty distinguishing between brands and descriptions. However, the even lower performance on the spoken version could be due to misrecognized brands; for example, "don julio tortillas" was incorrectly recognized as "on whole wheat tortillas." In the future, we may merge the brand and description categories since they cause confusion.

1) CNN Analysis: Often it is challenging to determine why neural networks perform so well; thus, we analyzed what each of the CNN's 64 filters in the single layer model learned. Figs. 5 and 6 show the 10 words with the highest activation in two of the CNN filters. We see that filter 52 fires on words related to potatoes, whereas filter 1 identifies fish.

In addition, we analyzed the activations for each token in a meal description. Fig. 7 shows the CNN filters with the highest

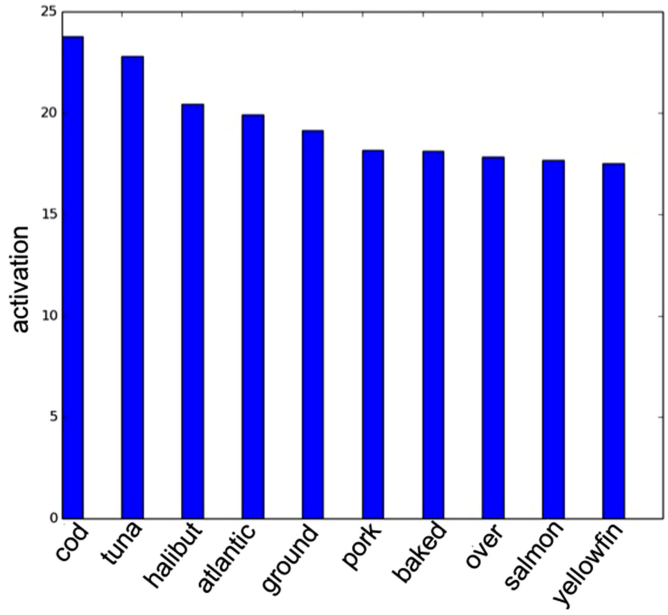

Fig. 6. The 10 words with the highest response to filter 1 of the CNN appear to mostly identify different types of fish.

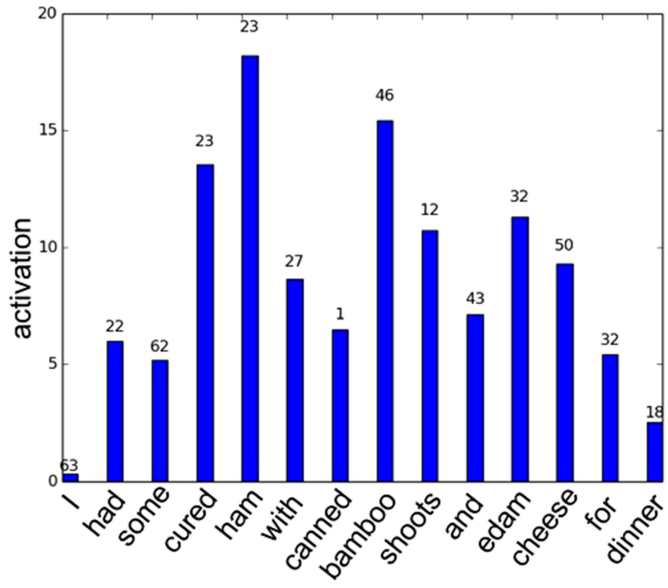

Fig. 7. The CNN filters with the highest activations for each token in a meal, where filter 23 fires on "cured ham" and filter 46 responds to "bamboo."

response to each token in the meal description "I had some cured ham with canned bamboo shoots and edam cheese for dinner." We observe that filter 23 learns "cured ham," which also has the highest activation of the whole sentence. See Appendix A for additional examples.

2) Tagging Performance With Increasing Data: We evaluated the semantic tagging performance on expert-labeled test data of the CRF with the highest performing feature set, as well as the best CNN with three stacked layers, as a function of the amount of training data (see Fig. 8). Increasing the amount of training data from 10,000 to 22,000 meal descriptions improved the semantic tagging performance of both models. It is

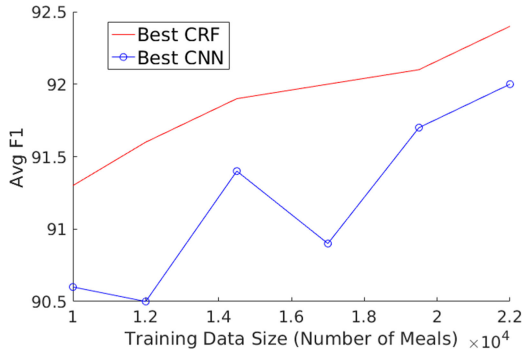

Fig. 8. Semantic tagging average F1 score over increasing amounts of data, for the best CRF versus the best CNN.

TABLE V
PERFORMANCE OF CRF+TBL ON THE SEGMENTATION TASK USING THREE DIFFERENT LABEL REPRESENTATIONS

Type	Accuracy	Precision	Recall	F1
IOB	86.9	63.3	60.6	61.9
IOE	**87.2**	63.2	**65.3**	**64.2**
IOBES	83.0	**64.0**	62.7	63.4

interesting to note that the CRF handles smaller data sets better than the CNN, whereas the CNN's performance fluctuates more and seems to have greater performance improvements with increased data; we expect the CNN to show even larger gains when trained on more data. Although the models perform similarly well, the CNN trains much faster (i.e., 37.4 seconds versus two hours and eight minutes for the CRF) and requires no feature engineering, which are both valuable contributions.

B. Property Association Experiments

Given the predicted semantic tags, we are still left with the remaining task of determining which properties are associated with which foods. Here we explore the performance of six different segmental approaches to food-property association, using both predicted and oracle semantic tags; we then analyze the non-segmental classifiers and show that the combination of the two outperforms each individually. Finally, we demonstrate that a neural network performs the best using predicted tags.

1) Segmental Experiments: In our prior work [42], we showed that the CRF model performed best out of the four segmental approaches, achieving a token-level accuracy of 86.9% and a phrase-level F1 score of 61.9 on a smaller data set of 8,000 meal descriptions. In addition to the IOB labeling scheme described in Section V-B1, there are two other representations for chunking: IOE (where the "end" token is represented by E) and IOBES (where single tokens are represented by S). We experimented with these other class types using the TBL algorithm on top of the CRF classifier. In Table V, we see that IOE has the highest F1 score, though the accuracies are not significantly different ($p < 0.01$).

Finally, we conducted oracle experiments in order to observe how well the models performed on the segmenting task when using the gold standard AMT labels, rather than the semi-

TABLE VI
FOOD-PROPERTY ASSOCIATION EXPERIMENTS WITH SIX SEGMENTAL METHODS, WHERE EXPERIMENTS WITH SEMI-CRF PREDICTIONS ARE COMPARED TO THOSE USING AMT GOLD STANDARD LABELS

Approach	F1 (Predicted)	F1 (Oracle)
Simple Rule	48.0	70.9
Simple + TBL	57.5	77.9
MM	49.2	72.0
MM + TBL	56.4	74.2
CRF	**61.9**	**78.3**
CRF + TBL	61.9	78.2

TABLE VII
PERFORMANCE ON THE FOOD-PROPERTY ASSOCIATION TASK USING THE PRIOR APPROACH OF IOE SEGMENTING WITH THE CRF, THE RANDOM FOREST CLASSIFICATION METHOD, AND THE UNION

Model	Precision	Recall	F1
Segmenting (Oracle)	87.9	83.9	85.9
Classifying (Oracle)	96.2	96.2	96.2
Combined (Oracle)	**96.5**	**96.5**	**96.5**
Segmenting (Predicted)	**86.2**	81.0	83.5
Classifying (Predicted)	84.7	87.9	86.3
Combined (Predicted)	84.9	**88.2**	**86.5**

Oracle experiments use true tags; predicted use CRF-predicted tags.

Markov CRF predictions (we were using the semi-Markov CRF at the time of the experiments). Since the semi-CRF labeling errors are compounded when fed into the segmenting task, we investigated how much the segmenter improved when given correct labels. As shown in Table VI, the F1 scores for all six methods in the oracle experiments were significantly higher than those from the non-oracle experiments. Again, TBL significantly improved upon the simple rule and the Markov model, but not the CRF. Therefore, the CRF alone had the overall best accuracy and F1 score.

2) Non-Segmental Experiments: To compare the performance of the random forest classifier to that of IOE chunking (i.e., segmentation with a CRF), we added IOE labels as additional features for both oracle and non-oracle experiments (see Table VII). These results show that using a random forest classifier yields a significantly higher F1 score than the CRF ($p < 0.01$), when evaluated on property tokens. For the CRF method, the number of gold property tokens with associated foods is greater than the number of property tokens with predicted foods, which indicates that some properties were missed in the IOE chunking scheme and therefore were not assigned any foods.

Our investigation demonstrates that the IOE labels from the CRF are complementary to the classification approach. As shown in the last row of both sections in Table VII, the addition of IOE labels as new features in the random forest classifier improved classification performance for both oracle and non-oracle experiments.

An error analysis revealed scenarios in which the non-segmental approach improved over the segmental approach, as well as those in which the combination outperformed both individual approaches. For example, in the meal "1.5 Cups of Honey Nut Cheerios 1/2 Cup Skim Milk," the segmental method incorrectly predicts "Nut" as the food which "1.5 Cups" and "Honey" describe, whereas the non-segmental method correctly selects "Cheerios" as the food associated with these properties. Clearly, the segmental approach has difficulty handling incorrectly predicted semantic tags (i.e., "Nut"), whereas the non-segmental approach can handle this by directly predicting the corresponding food (i.e., "Cheerios") despite it appearing further away from the properties. In the meal description, "veggie patty, one 365 Whole Foods piece of...naan bread...," the non-segmental approach mistakenly predicts "patty" as the food associated with the brand "365," rather than "bread." However, in this case, the combined method is able to leverage the correct association identified by the segmental approach in order to fix the mistake.

TABLE VIII
PERFORMANCE ON FOOD-PROPERTY ASSOCIATION USING THE RANDOM FOREST AND NEURAL NETWORK, ON TEXT AND SPEECH

Model	Precision	Recall	F1
Classif. (Oracle)	**94.2**	**94.2**	**94.2**
NN (Oracle)	93.7	93.7	93.7
Speech: Classif. (Oracle)	**99.0**	**99.0**	**99.0**
Speech: NN (Oracle)	98.8	98.8	98.8
Classif. (Predicted)	86.1	83.4	84.7
NN (Predicted)	**89.2**	**86.2**	**87.7**
Speech: Classif. (Predicted)	84.2	90.2	87.1
Speech: NN (Predicted)	**86.7**	**93.1**	**89.8**

Finally, we addressed the research question of whether we can perform similarly well with a neural network. We trained a feed-forward network on the full set of 22,000 annotated meal descriptions, with a split of 90% for training and 10% for testing. Table VIII shows that although the random forest classifier does better on oracle experiments, the neural network outperforms the simple classifier when using predicted semantic tags, which is what the live system uses at test time. As expected, the performance is significantly better in the oracle experiments than when using predicted tags, where $p < 0.01$. The similar performance of both methods on the speech corpus indicates that using speech did not greatly impact performance. The performance is actually higher on speech for the oracle experiments, since the test set is much smaller (i.e., 239 spoken utterances versus 2,163 written meal logs). The decline in performance on the predicted experiments is likely due to semantic tagging errors.

C. System Evaluation

In order to evaluate the system's overall performance on people, we launched an AMT task where Turkers rated how well the system performed on three separate tasks: semantic tagging, quantity matching, and correctly identifying USDA (Nutrient Database for Standard Reference)[3] hits for matching foods. We asked Turkers to record two meal descriptions each and to interact with the system by revising the quantities and selecting a single USDA hit:

[3]http://ndb.nal.usda.gov/ndb/search

Please record or type two meals (e.g., what you ate for breakfast, lunch, dinner, or snack today or yesterday) using as much detail and accuracy as possible in the text box provided and click enter on your keyboard when you are done. Be creative - we will not accept repeat answers. Try to include as much additional information as you remember, such as brand names and quantities. Note that the system requires using Chrome. Please interact with the system to narrow down the USDA hits to one food and play with the quantities. Then **check the boxes** *in the right-most column if the labels are correct, if the quantity is correct, if the final USDA hit correctly matches the food you actually ate, and if the corresponding images are correct. If you encounter any errors or have feedback from your experience using the system, please let us know! Examples: I had a hard boiled egg, a whole wheat english muffin, and one tablespoon of peanut butter. For lunch I ate a sauteed onion, 3 ounces of chicken breast, and one cup of mixed vegetables.*

Using AMT for evaluation enabled us to quickly test the system on many more people than we could evaluate in traditional user studies. Research [41] has also demonstrated that Turker samples are more diverse than samples used in traditional user studies. However, the downside of crowdsourcing the system evaluation online is that there is always the risk of noise in AMT data, and we do not have the benefit of observing real users in person as they use the system. We plan to conduct a pilot study with patients of Tufts University nutritionists to address these limitations.

The results from 437 meal descriptions containing a total of 975 food concepts indicated that 83% of semantic tags were correct, 78% of the quantities were correct, and 71% of the USDA hits were correct matches. There were only 34 insertions (i.e., a non-food token labeled as food) and 96 substitutions (i.e., a food token labeled as non-food).

The system did not use the best models (due to difficulty porting Python experiments to the system in Java) and thus had a lower semantic tagging performance of 83.5 on the spoken test data, as well as a food-property association performance of 83.4 on spoken data. Since the system was built initially in Java, but we ran all our most recent experiments in Python, we would need to re-implement all the models in Java, including the neural networks and word embedding and prototype similarity features for the CRFs. The alternative, which we are pursuing in our current work, is to set up a new Python Flask server[4] which uses Docker[5] to install the dependencies (e.g., SpaCy for tokenization and Keras for neural networks). Thus, these engineering challenges precluded inclusion of the best system in the evaluation on AMT of the live system. We instead launched an older Java implementation, which used a Mallet[6] CRF trained only on basic n-gram and part-of-speech tag features for semantic tagging, and CRF++[7] with IOE labeling for food-property association.

VII. Conclusion

In this paper, we have examined the language understanding component of a novel food logging system that allows obesity patients to monitor their caloric and nutrient intake more easily and efficiently than existing self-assessment methods. The methods presented here focused on the data collection and language understanding methods for two components: semantic tagging and food-property association. Our primary contributions are as follows:

1) We showed that the CRF model is a viable method for semantic tagging of written and spoken meal descriptions. Our evaluation of the system's performance on Mechanical Turk indicated that semantic tagging in the deployed system is reasonably accurate when tested by Turkers.
2) We verified that incorporating word vector features inspired by distributional semantics theory improves a standard CRF classifier's semantic tagging performance.
3) We demonstrated that a deep CNN with no manually designed features outperforms the best CRF for semantic tagging on spoken data; visual analysis shows the CNN filters learn to identify meaningful categories of words.
4) We found that segmental and non-segmental methods for food-property association are complementary.

As part of our ongoing work, we are exploring character-based CNNs [76]–[78] for generating word embeddings that are less sensitive to misspellings and other inconsistencies in written meal logs, since users often type foods and brands incorrectly (e.g., "kellogs" vs. "Kellogg's"). In addition, we will investigate neural conversational agents for discussing health and nutrition with users. Finally, we plan to use multi-task learning [20] for jointly training a neural network model to do both semantic tagging and food-property association.

Appendix

In Figs. 9 and 10, we show two additional learned CNN filters (see Section VI-A1) with the words that yield the highest activations.

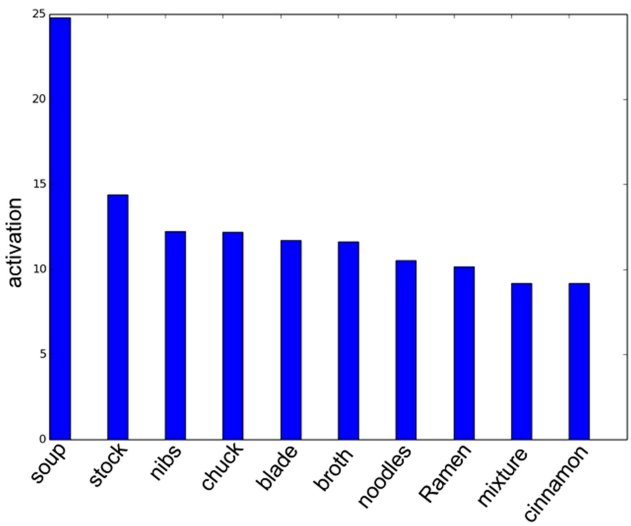

Fig. 9. The top 10 words related to soup that have the highest response to filter 10 of the CNN.

[4]http://flask.pocoo.org/
[5]https://www.docker.com/
[6]http://mallet.cs.umass.edu/
[7]https://taku910.github.io/crfpp/

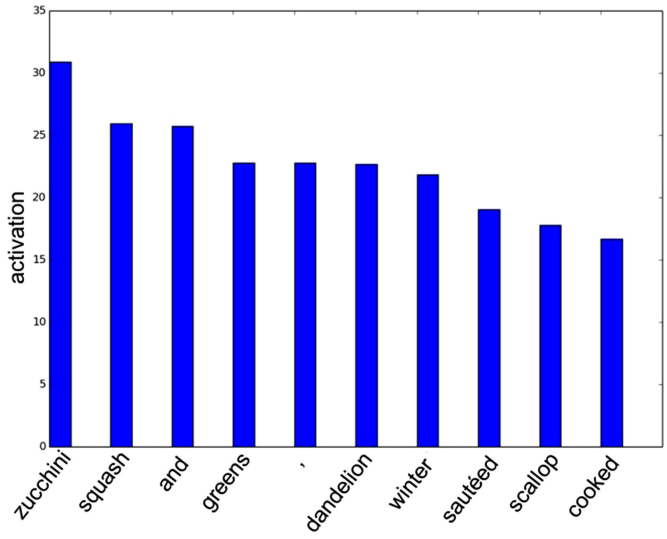

Fig. 10. The top 10 words, mostly involving vegetables, that have the highest activation in response to CNN filter 28.

ACKNOWLEDGMENT

M. Price built the nutrition speech recognizer [46], R. Naphtal [79] implemented the nutrition database lookup, P. Saylor [47] helped with the web audio interface, and S. Cyphers contributed to the trie matching.

REFERENCES

[1] A. Tsai, D. Williamson, and H. Glick, "Direct medical cost of overweight and obesity in the USA: A quantitative systematic review," *Obesity Rev.*, vol. 12, no. 1, pp. 50–61, 2011.

[2] 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 Rev.*, vol. 29, no. 1, pp. 6–28, 2007.

[3] W. H. Organization, *Obesity: Preventing and Managing the Global Epidemic*. Geneva, Switzerland: World Health Organization, 2000, no. 894.

[4] M. Korpusik, "Spoken language understanding in a nutrition dialogue system," Master's thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, 2015.

[5] M. Korpusik, R. Naphtal, N. Schmidt, S. Cyphers, and J. Glass, "Nutrition system demonstration," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014.

[6] C. Hemphill *et al.*, "The ATIS spoken language systems pilot corpus," in *Proc. DARPA Speech Natural Lang. Workshop*, 1990, pp. 96–101.

[7] D. Dahl *et al.*, "Expanding the scope of the ATIS task: The ATIS-3 corpus," in *Proc. Workshop Human Lang. Technol.*, Association for Computational Linguistics, 1994, pp. 43–48.

[8] Y. He and S. Young, "A data-driven spoken language understanding system," in *Proc. 2003 IEEE Workshop Automat. Speech Recog. Understanding*, IEEE, 2003, pp. 583–588.

[9] Y. Wang, A. Acero, M. Mahajan, and J. Lee, "Combining statistical and knowledge-based spoken language understanding in conditional models," in *Proc. COLING/ACL Main Conf. Poster Sessions*, Association for Computational Linguistics, 2006, pp. 882–889.

[10] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proc. 8th Int. Conf. Spoken Lang. Process.*, 2007, pp. 1605–1608.

[11] I. Meza-Ruiz, S. Riedel, and O. Lemon, "Accurate statistical spoken language understanding from limited development resources," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, IEEE, 2008, pp. 5021–5024.

[12] S. Heintze, T. Baumann, and D. Schlangen, "Comparing local and sequential models for statistical incremental natural language understanding," in *Proc. 11th Annu. Meeting Special Interest Group Discourse Dialogue*, Association for Computational Linguistics, 2010, pp. 9–16.

[13] G. Tur, D. Hakkani-Tür, L. Heck, and S. Parthasarathy, "Sentence simplification for spoken language understanding," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, IEEE, 2011, pp. 5628–5631.

[14] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?" in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2010, pp. 19–24.

[15] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. 14th Int. Conf. Spoken Lang. Process.*, 2013, pp. 3771–3775.

[16] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2014, pp. 189–194.

[17] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[18] D. Lin and X. Wu, "Phrase clustering for discriminative learning," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. AFNLP: Volume 2*, Association for Computational Linguistics, 2009, pp. 1030–1038.

[19] A. Passos, V. Kumar, and A. McCallum, "Lexicon infused phrase embeddings for named entity resolution," in *Proc. 18th Conf. Comput. Lang. Learn.*, Baltimore, MD, USA, Jun. 2014, pp. 78–86.

[20] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[21] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," arXiv:1508.01991, 2015.

[22] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, San Diego, CA, USA, Jun. 2016, pp. 260–270.

[23] S. Seneff and J. Polifroni, "Dialogue management in the Mercury flight reservation system," in *Proc. ANLP/NAACL Workshop Conversational Syst.—Volume 3*. Association for Computational Linguistics, 2000, pp. 11–16.

[24] B. Thomson, *Statistical Methods for Spoken Dialogue Management*. New York, NY, USA: Springer, 2013.

[25] J. Liu, P. Pasupat, S. Cyphers, and J. Glass, "Asgard: A portable architecture for multilingual dialogue systems," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8386–8390.

[26] S. Young *et al.*, "The hidden information state model: A practical framework for POMDP-based spoken dialogue management," *Comput. Speech Lang.*, vol. 24, no. 2, pp. 150–174, 2010.

[27] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural nnetworks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[28] O. Vinyals and Q. Le, "A neural conversational model," arXiv:1506.05869, 2015.

[29] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, 2015.

[30] A. Sordoni *et al.*, "A neural network approach to context-sensitive generation of conversational responses," *Human Lang. Technologies: 2015 Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Denver, CO, USA, May/Jun. 2015, pp. 196–205.

[31] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, San Diego, CA, USA, Jun. 2016, pp. 110–119.

[32] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 994–1003.

[33] P. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, 2010.

[34] M. Baroni, G. Dinu, and G. Kruszewski, "Dont count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 238–247.

[35] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2013, pp. 746–751.

[36] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang.*, vol. 12, 2014.

[37] R. Al-Rfou, B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," in *Proc. 17th Conf. Comput. Natural Lang. Learn.*, Sofia, Bulgaria, Aug. 2013, pp. 183–192.
[38] Y. Chen and A. Rudnicky, "Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014.
[39] A. Celikyilmaz, D. Hakkani-Tur, P. Pasupat, and R. Sarikaya, "Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems," *2015 AAAI Spring Symp.: Knowledge Representation Reasoning: Integrating Symbolic Neural Approaches.*
[40] R. Manuvinakurike, C. Kennington, D. DeVault, and D. Schlangen, "Real-time understanding of complex discriminative scene descriptions," in *Proc. 17th Annu. SIGdial Meeting Discourse Dialogue*, 2016, pp. 232–241.
[41] M. Buhrmester, T. Kwang, and S. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives Psychological Sci.*, vol. 6, no. 1, pp. 3–5, 2011.
[42] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass, "Data collection and language understanding of food descriptions," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014.
[43] A. Viera *et al.*, "Understanding interobserver agreement: The kappa statistic," *Family Med.*, vol. 37, no. 5, pp. 360–363, 2005.
[44] D. Knuth, J. Morris, and V. Pratt, "Fast pattern matching in strings," *SIAM J. Comput.*, vol. 6, no. 2, pp. 323–350, 1977.
[45] D. Conway, "An algorithmic approach to English pluralization," in *Proc. 2nd Annu. Perl Conf. C. Salzenberg*. San Jose, CA, O'Reilly, 1998.
[46] M. Korpusik, C. Huang, M. Price, and J. Glass, "Distributional semantics for understanding spoken meal descriptions," in *Proc. IEEE Conf. Acoustics, Speech Signal Process.*, 2016.
[47] P. Saylor, "Spoke: A framework for building speech-enabled websites," Master's thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, 2015.
[48] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE Workshop Automat. Speech Recog. Understanding*, Dec. 2011.
[49] C. Sutton *et al.*, "An introduction to conditional random fields," *Foundations Trends Mach. Learn.*, vol. 4, no. 4, pp. 267–373, 2012.
[50] H. Schütze, "Word space," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, vol. 5, pp. 895–902.
[51] G. Miller and W. Charles, "Contextual correlates of semantic similarity," *Lang. Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.
[52] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
[54] N. Okazaki, "CRFsuite: A fast implementation of conditional random fields (CRFs)," 2007. [Online]. Available: http://www.chokkan.org/software/crfsuite/
[55] G. Lakoff, *Women, Fire, and Dangerous Things*. Chicago, IL, USA: Univ. Chicago Press, 2008.
[56] J. Guo, W. Che, H. Wang, and T. Liu, "Revisiting embedding features for simple semi-supervised learning," in *Proc. Conf. Empirical Methods Natural Lang.*, 2014, pp. 110–120.
[57] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[58] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, 2003.
[59] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. 14th Int. Conf. Spoken Lang. Process.*, 2013, pp. 3771–3775.
[60] K. Yao, G. Zweig, M. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *Proc. 14th Int. Conf. Spoken Lang. Process.*, 2013, pp. 2524–2528.
[61] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, 2016, vol. 4, pp. 259–272.
[62] W. Yin, S. Ebert, and H. Schütze, "Attention-based convolutional neural network for machine comprehension," arXiv:1602.04341, 2016.
[63] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," arXiv:1606.01781, 2016.
[64] F. C., "keras," 2015. [Online]. Available: https://github.com/fchollet/keras
[65] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
[66] T. Lei *et al.*, "Denoising bodies to titles: Retrieving similar questions with recurrent convolutional models," arXiv:1512.05726, 2015.
[67] C. Bishop *et al.*, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, vol. 1.
[68] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," *Introduction Statist. Relational Learn.*, pp. 93–128, 2006.
[69] I. Hetherington, "The MIT finite-state transducer toolkit for speech and language processing," in *Proc. 8th Int. Conf. Spoken Lang. Process.*, 2004.
[70] R. Florian and G. Ngai, "Fast transformation-based learning toolkit," Johns Hopkins University, Baltimore, MD, USA, 2001.
[71] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[72] T. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2015, pp. 39–48.
[73] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, 1989, pp. 532–535.
[74] S. Gebhardt *et al.*, "USDA national nutrient database for standard reference, release 21," Beltsville, MD, USA: U.S. Dept. Agriculture, Agricultural Res. Service, Beltsville Human Nutrition Research Center, Nutrient Data Laboratory, 2008.
[75] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Syst. Demonstrations)*, 2014, pp. 55–60.
[76] Y. Kim, Y. Jernite, D. Sontag, and A. Rush, "Character-aware neural language models," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 2741–2749.
[77] C. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1818–1826.
[78] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[79] R. Naphtal, "Natural language processing based nutritional application," Master's thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambdridge, MA, 2015.

Mandy Korpusik (S'16) received the B.S. degree in electrical and computer engineering from Olin College of Engineering, Needham, MA, USA, in May 2013 and the S.M. degree in computer science at MIT, Cambridge, MA, USA, in June 2015. She is working toward the Ph.D. degree with Dr. J. Glass in Computer Science and Artificial Intelligence Laboratory, MIT. Her primary research interests include natural language processing and spoken language understanding, and she has previously worked on user intent detection, semantic tagging, and predicting user purchase behavior. She is currently working on a nutrition dialogue system that uses convolutional neural networks to map between natural language meal descriptions and USDA food database entries. She is also interested in encoder–decoder models for conversational agents, caption generation, and translation.

James Glass (F'14) is a Senior Research Scientist at MIT, Cambridge, MA, USA, where he leads the Spoken Language Systems Group in the Computer Science and Artificial Intelligence Laboratory. He is also a member of the Harvard-MIT Health Sciences and Technology Faculty. Since obtaining the S.M. and Ph.D. degrees at MIT in electrical engineering and computer science, his research has focused on automatic speech recognition, unsupervised speech processing, and spoken language understanding. He is a Fellow of the International Speech Communication Association, and is currently an Associate Editor for *Computer, Speech, and Language*.