DISTRIBUTIONAL SEMANTICS FOR UNDERSTANDING SPOKEN MEAL DESCRIPTIONS

*Mandy Korpusik, Calvin Huang, Michael Price, and James Glass**

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

ABSTRACT

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 discuss the technical approaches to understanding, including three methods for incorporating word vector features into conditional random field (CRF) models for semantic tagging, as well as classifiers for directly associating foods with properties. We report experiments on both text and spoken data from an in-domain speech recognizer. On text data, we show that the addition of word vector features significantly improves performance, achieving an F1 test score of 90.8 for semantic tagging and 86.3 for food-property association. On speech, the best model achieves an F1 test score of 87.5 for semantic tagging and 86.0 for association. Finally, we conduct an end-to-end system evaluation through a user study with human ratings of 83% semantic tagging accuracy.

Index Terms— CRF, Word vectors, Semantic tagging

1. INTRODUCTION

For many patients with obesity, diet tracking is often time-consuming and cumbersome, especially for hard-to-reach, low-literate populations [1, 2]. In an effort to improve obesity treatment and prevention techniques, we have begun developing a nutrition system that accurately and efficiently records food intake through speech and language technology. Existing applications for tracking nutrient and caloric intake, such as MyFitnessPal [3], require manually entering food items one at a time and selecting the correct match from a list of database entries, whereas our system enables automatic detection of food concepts through spoken language understanding.

The flow of the overall nutrition system is shown in Figure 1. After the user generates a meal description by typing or speaking (to a speech recognizer), 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) models for the language understanding tasks: semantic tagging (i.e., labeling tokens) and food-property association. The language understanding output is used for database lookup and image search before responding to the user.

In [5], we discussed the initial data collection and language understanding of our nutrition system prototype. In this paper, we incorporate distributional semantics into semantic tagging models, describe a new approach for associating foods with properties, build a domain-specific speech recognizer for evaluation on spoken data, and evaluate the system in a user study. Specifically, our contributions are as follows:

*This research was sponsored by a grant from Quanta Computing, Inc., and by the NIH.

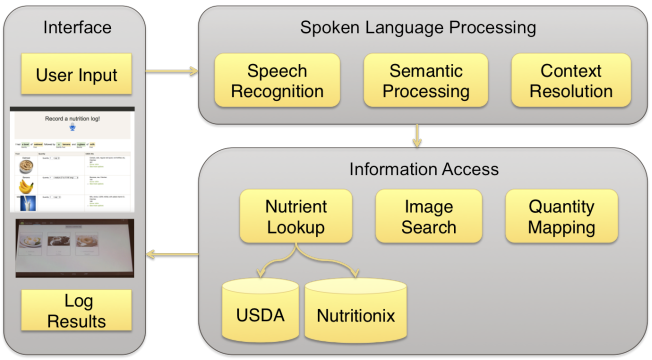

Fig. 1. A diagram of the flow of the nutrition system [4].

- We demonstrate a significant improvement in semantic tagging performance by adding word embedding features to a classifier. We explore features for both the raw vector values and similarity to prototype words from each category. In addition, improving semantic tagging performance benefits the subsequent task of associating foods with properties.
- We build a nutrition speech recognizer for evaluating the language understanding models on spoken data. We also evaluate the system in a user study on Amazon Mechanical Turk.

Related work [6] explored semantic parsing for cooking recipes; however, for their concept identification, they utilized logistic regression, whereas we employed a CRF for semantic tagging. In addition, they focused on procedural text (i.e., sets of instructions), which required building a directed acyclic flow graph from recipe text; in our work, for food-property association we directly predicted relations using a random forest classifier.

Recent work [7, 8, 9] has shown that using word embeddings as features in classifiers can improve natural language processing performance in a variety of tasks, such as part-of-speech tagging in multiple languages [10], enriching spoken queries in dialogue systems [11], and semantic tagging [12]. We investigated three approaches for incorporating word embedding features into a CRF semantic tagging model: using dense embedding values directly, measuring the cosine distance between tokens and "prototypes" (i.e., words most representative of a category, such as "bread" for foods), and clustering vectors.

In the remainder of this paper, we begin by presenting the technical approach to the two language understanding tasks: semantic tagging and food-property association. Section 3 summarizes the text and speech corpora, experimental results, and system evaluation. Section 4 discusses the results, and Section 5 concludes.

2. LANGUAGE UNDERSTANDING

The language understanding component of the system is composed of two tasks: tagging each token in a spoken meal description as a food, brand, quantity, or description; and assigning properties to foods. For example, in the meal shown in Figure 2, the tokens "cereal" and "milk" are tagged as foods, whereas "a bowl" and "two cups" are quantities. Subsequently, "a bowl" is assigned to "cereal," while "two cups" is associated with "milk."

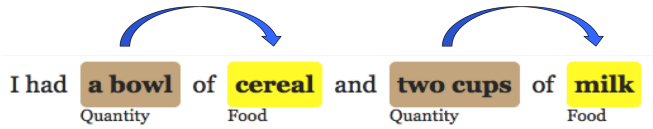

Fig. 2. A depiction of the two language understanding tasks.

2.1. Semantic Tagging

In this section, we discuss the first language understanding task, semantic tagging, where we incorporate word embeddings as features in a CRF model. In prior work [5], we compared a semi-Markov conditional random field (semi-CRF) to a standard CRF baseline, with which we predicted a vector of output food and property labels $\vec{y} = \{y_0, y_1, ..., y_T\}$ corresponding to a set of input feature vectors for each token (or segment of tokens) in a meal description $\vec{x} = \{\vec{x_0}, \vec{x_1}, ..., \vec{x_T}\}$. The baseline features included n-grams, part-of-speech (POS) tags, and presence in a food or brand lexicon.

According to distributional semantics theory [13, 14], words with similar meanings have similar vector representations, so we explored using neural network-trained vectors as CRF tagging features to account for semantics. We used three methods for incorporating such vectors: dense embedding values, binary and raw similarity values between tokens and prototypes [15], and clusters.

A popular method for learning word embeddings is Mikolov's Skip-gram model [16], released as the word2vec toolkit [17], which learns word vector representations that best predict the context surrounding a word. Given training words $w_1, w_2, ..., w_T$, the objective is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t) \quad (1)$$

where c is the size of the context window around center word w_t. The probability $p(w_{t+j}|w_t)$ is estimated by the softmax

$$p(w_O|w_I) = \frac{\exp\left(v_{w_O}^{\top} v_{w_I}\right)}{\sum_{w'_O=1}^{W} \exp\left(v_{w'_O}^{\top} v_{w_I}\right)} \quad (2)$$

where v_{w_I} and v_{w_O} are the input and output vectors of w, and W is the vocabulary size. In our experiments, we trained the vectors with the continuous bag-of-words (CBOW) approach, which predicts the current word based on the context [18].

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[1]). For these experiments, we employed the CRFsuite [19] implementation rather than CRF++ (although performance was similar) for two reasons: faster running time and the ability to use vector float values directly.

In addition to using the continuous, dense embeddings as features in our models, we explored a distributional prototype method for discretizing the embedding features: representing each label category with a prototype word (see Table 1) and using the similarity between a token and prototypes as features [15]. 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). For each category, 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.

Label	Prototypes
Food	"water," "milk," "sauce," "coffee," "tea"
Brand	"Kraft," "Trader," "Great," "Kroger," "Joe's"
Quantity	"cup," "two," "glass," "oz," "one"
Description	"white," "green," "peanut," "black," "whole"

Table 1. Top five prototypes for each category (except Other).

Finally, we investigated clustering as an alternative to the distributional prototype method for discretizing continuous word vectors. Using word2vec's k-means clustering algorithm ($k = 500$), we added a feature for each token's cluster.

2.2. Property Association

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 Figure 2). There are two alternative approaches for accomplishing this food-property association task: segmenting the meal description tokens into food chunks (each with a food item and its properties), and predicting the most likely food for each property.

As an alternative to the four segmenting methods we explored in [5], in this work we trained a classifier for assigning properties to foods. This approach indirectly incorporated word embeddings into the food-property association task by using the predicted semantic tags from the CRF trained on word vectors. We compared oracle experiments with gold standard tags to experiments using predicted tags, where we used the model with the best feature combination.

One drawback to using the segmenting 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 IOE format (i.e., I indicates inside a chunk, O outside, and E the end) requires making assumptions where some information (e.g., long-range dependencies)

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

is omitted. Thus, we investigated an alternative method for food-property association where we trained a classifier to directly predict which food a property describes.

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 yet 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 a single hard label.

The classification is done using six 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, the property's entity type if it is a named entity, and the dependency relation between the property and food token if the food is the property token's head in the dependency parse tree of the meal log. We explored three different classifiers, using the Scikit-learn toolkit's implementation for Python [20]: 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. Performance was evaluated using precision, recall, and F1 scores for property tokens only. In the oracle experiments, since we used the gold standard semantic tags, the number of actual property tokens equals the number of predicted property tokens, so the precision, recall, and F1 scores are all equivalent.

3. EXPERIMENTS

In order to evaluate our semantic tagging and food-property association models, we collected and annotated a set of text data, as well as spoken data. To prepare the speech data, we built an in-domain speech recognizer from audio recordings of meal descriptions, and labeled the recognizer's output on Amazon Mechanical Turk (AMT). In addition, we evaluated the end-to-end system on AMT.

3.1. Text and Speech Corpus Descriptions

We evaluated our models on a data set of 10,000 textual meal descriptions collected and annotated via AMT [21]. We collected and labeled 2,000 food logs each of breakfast, lunch, dinner, and snacks. We also launched an AMT task with the deployed nutrition system, in which we asked 500 Turkers to record four meal descriptions, yielding 2,000 additional meal descriptions. The data were tokenized on spaces, and if one of the resulting strings began or ended with a punctuation mark, we further split the token on the punctuation. The data were divided into 90% training and 10% testing.

The experiments presented in prior work [5] relied upon written, rather than spoken, data. To address this limitation, we collected a corpus of spoken meal descriptions, and created a nutrition speech recognizer. We collected the speech data via AMT [22], where we asked Turkers to record 10 meal descriptions. 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. Using Kaldi [23], we trained a 256 node, 6 layer, deep neural network (DNN) acoustic model and a trigram language model on 40,000 written meal diaries. 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 [24] for subsequent understanding evaluation.

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

3.2. Semantic Tagging

As the baseline, we used n-gram features (as-is and lowercase), POS tags [25], and presence in USDA food and brand lexicons [26]. To improve upon the baseline, first we added dense embedding features (row three in Table 2). Next, we incorporated distributional prototype similarity features (fourth row in Table 2). Since many high scoring prototypes did not appear in the corpus, we selected the next best prototype with a vector representation. We did not include prototypes for the "None" category because this reduced performance.

Model	Food	Brand	Num	Descr	None	Avg
No CRF	85.7	73.6	89.3	75.7	92.6	83.4
Baseline	94.3	81.4	91.9	88.6	95.1	90.2
+Dense	94.5	81.5	91.9	88.7	95.1	90.3
+Proto	94.9	82.4	**91.9**	89.0	**95.3**	90.7
+Shape	94.9	82.8	91.7	89.1	95.1	90.7
+Cluster	**95.0**	**82.8**	91.7	**89.1**	95.1	**90.8**
+Speech	95.9	68.5	91.6	87.3	94.1	87.5

Table 2. CRFsuite F1 scores per label in the semantic tagging task with incrementally complex feature sets: baseline n-grams, POS tags, and lexical features; dense embeddings; raw prototype similarities; shape; and clusters. For comparison, the top row predicts tags using the most frequent tag in the training data for a token (or "None" if unseen during training). The last row shows evaluation on the speech test corpus for the full feature set.

We experimented with two similarity features: a binary number indicating whether or not the similarity was below a threshold δ determined through cross-validation (i.e., 0.25 for brands and 0.3 for the other categories) and the raw value of the similarity. The raw similarity value features improved upon the baseline combined with raw word vector values, but the binary similarity did not yield further improvement. This might be because the CRF is able to determine the relative importance of each similarity feature, whereas the binary similarity weights the different prototypes' similarities equally.

The shape feature, which indicated whether or not the token was in titlecase, lowercase, uppercase, a number, or a piece of punctuation, improved upon the baseline and prototype features (row five in Table 2). Finally, k-means clusters (500 classes) yielded the highest average F1 score of 90.8; the improvement of all combined features was significant (McNemar), relative to the baseline, where $p < 0.05$. For comparison, the performance of the best semantic tagging model on the speech corpus is shown in the last row of Table 2.

3.3. Property Association

The highest scoring association model on the text corpus is the random forest classifier (see Table 3). As expected, the performance is significantly better in the oracle experiments than when using predicted tags, where $p < 0.01$.The similar performance of the random forest on the speech corpus indicates that using speech did not greatly impact performance.

Model	Prec.	Recall	F1
Naive Bayes (Oracle)	94.6	94.6	94.6
Logistic Regression (Oracle)	95.2	95.2	95.2
Random Forest (Oracle)	**96.2**	**96.2**	**96.2**
Naive Bayes (Predicted)	84.1	87.3	85.7
Logistic Regression (Predicted)	84.2	87.4	85.7
Random Forest (Predicted)	**84.7**	**87.9**	**86.3**
Speech: Random Forest (Predicted)	82.4	89.8	86.0
Speech: Random Forest (Oracle)	**98.5**	**98.5**	**98.5**

Table 3. Food-property association with three classifiers, for gold standard tags (i.e., oracle) and predicted tags. The performance on the speech corpus is shown in the last two rows.

To compare the performance of the classification approach to that of IOE chunking (i.e., segmentation), we added IOE labels as additional features for both oracle and non-oracle experiments (see Table 4). These results show that the new association approach 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.

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**

Table 4. Performance on the food-property association task using the prior approach of IOE segmenting with the CRF, the new random forest classification method, and the union.

We also investigated whether the IOE labels from the CRF were complementary to the food-property classification approach by incorporating the predicted IOE labels as new features in the random forest classifier. As shown in the last row of both sections in Table 4, the addition of IOE labels improved classification performance for both oracle and non-oracle experiments. The union performed significantly better than the CRF segmenter alone, where $p < 0.01$.

3.4. System Evaluation

In order to evaluate the system's overall performance on real users, 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. 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.

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

4. DISCUSSION

We measured the reliability of the textual data annotations by calculating the inter-annotator agreement among Turkers. The kappa score [27] for the food labeling task is 0.77, which indicates substantial agreement, whereas the kappa score for the property labeling task is 0.41. The property labeling task was more challenging, since there were three possible categories instead of one; in addition, distinguishing between brands and descriptions was difficult.

For the semantic tagging experiments, we explored training vectors on the Google News corpus, as well as on domain-specific nutrition data; however, the vectors trained on Google News performed best because the data is much larger than the nutrition data set. In the future, we may expand the nutrition data for training domain-specific vectors by extracting recipes from the web. When selecting the best word vector features for tagging, we also found that using a unique feature for each prototype's similarity, as opposed to averaging all the similarities, improved performance. This could be due to differences in meaning among the prototypes within a category. For example, drinks such as "juice" are considered foods, as is "bread;" however, "tea" might be similar to "juice," but not to "bread."

The average semantic tagging F1 test score on spoken data (87.5) is somewhat lower than the best score displayed in Table 2 on text data (90.8), which is probably due to the difference in test data size (1,000 vs. 251 diaries); the association performance is not as affected by the use of speech data. The low performance on semantic tagging of brands 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. In the future, we may merge the brand and description categories, since they ultimately serve the same purpose in the nutrition database lookup.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we explored three approaches for incorporating word embedding features into CRFs for the semantic tagging task of a nutrition system. The best feature set consisted of baseline, shape, and cluster features combined with raw word embedding values and cosine similarity scores to prototype vectors, yielding an average F1 score of 90.8. For the food-property association task, we investigated applying classifiers to directly predict the food that a property describes; the random forest classifier achieved an F1 score of 86.3 with predicted tags. We evaluated the models on spoken data from a nutrition speech recognizer. Finally, when we evaluated the system prototype in a user study, semantic tagging was 83% accurate.

In the future, we plan to explore multiple-sense embeddings [28], since many words have several meanings, and use letter trigram vectors [29] to handle unknown words. Finally, we plan to investigate the use of recurrent neural networks (RNNs) [30, 31], recurrent CRFs [32], long short-term memory (LSTM) [33], or recursive NNs [34, 35, 36] as alternatives to the CRF.

Acknowledgements

Rachael Naphtal helped with the database lookup, and Patricia Saylor helped with the web audio interface.

6. 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] World Health Organization, *Obesity: Preventing and Managing the Global Epidemic*, Number 894. World Health Organization, 2000.

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

[4] M. Korpusik, R. Naphtal, N. Schmidt, S. Cyphers, and J. Glass, "Nutrition system demonstration," *Proc. SLT*, 2014.

[5] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass, "Data collection and language understanding of food descriptions," *Proc. SLT*, 2014.

[6] H. Maeta, T. Sasada, and S. Mori, "A framework for procedural text understanding," in *Proc. IWPT*, 2015, pp. 50–60.

[7] M. Baroni, G. Dinu, and G. Kruszewski, "Dont count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. ACL*, 2014, vol. 1, pp. 238–247.

[8] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations.," in *HLT-NAACL*, 2013, pp. 746–751.

[9] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," *Proc. EMNLP*, vol. 12, 2014.

[10] R. Al-Rfou, B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," *arXiv preprint arXiv:1307.1662*, 2013.

[11] Y. Chen and A. Rudnicky, "Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings," *Proc. SLT*, 2014.

[12] A. Celikyilmaz, D. Hakkani-Tur, P. Pasupat, and R. Sarikaya, "Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems," *genre*, 2010.

[13] H. Schütze, "Word space," in *Advances in Neural Information Processing Systems 5*. Citeseer, 1993.

[14] G. Miller and W. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.

[15] J. Guo, W. Che, H. Wang, and T. Liu, "Revisiting embedding features for simple semi-supervised learning," in *Proc. EMNLP*, 2014, pp. 110–120.

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

[17] T. Mikolov, K. Chen, and J. Dean, "word2vec (2013)," .

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[19] Naoaki Okazaki, "CRFsuite: A fast implementation of conditional random fields (CRFs)," 2007.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] I. McGraw, S. Cyphers, P. Pasupat, J. Liu, and J. Glass, "Automating crowd-supervised learning for spoken language systems," in *Proc. INTERSPEECH*, 2012.

[22] P. Saylor, "Spoke: A framework for building speech-enabled websites," M.S. thesis, Massachusetts Institute of Technology, 2015.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, Dec. 2011.

[24] M. Korpusik, "Spoken language understanding in a nutrition dialogue system," M.S. thesis, Massachusetts Institute of Technology, 2015.

[25] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. ACL*, 2014, pp. 55–60.

[26] S. Gebhardt, L. Lemar, D. Haytowitz, P. Pehrsson, M. Nickle, B. Showell, R. Thomas, J. Exler, and J. Holden, "USDA national nutrient database for standard reference, release 21," 2008.

[27] A. Viera, J. Garrett, et al., "Understanding interobserver agreement: The kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005.

[28] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient nonparametric estimation of multiple embeddings per word in vector space," in *Proc. EMNLP*, 2014.

[29] W. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proc. ACL*, 2014.

[30] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. INTERSPEECH*, 2013, pp. 3771–3775.

[31] K. Yao, G. Zweig, M. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *Proc. INTERSPEECH*, 2013, pp. 2524–2528.

[32] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proc. ICASSP*. IEEE, 2014, pp. 4077–4081.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*. Citeseer, 2013, vol. 1631, p. 1642.

[35] D. Guo, G. Tur, W. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," *Proc. SLT*, 2014.

[36] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. III, "A neural network for factoid question answering over paragraphs," in *Proc. EMNLP*, 2014, pp. 633–644.