DATA COLLECTION AND LANGUAGE UNDERSTANDING OF FOOD DESCRIPTIONS

Mandy Korpusik, Nicole Schmidt, Jennifer Drexler, Scott Cyphers, and James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{korpusik, nicole16, jdrexler, cyphers, glass}@csail.mit.edu

ABSTRACT

This paper presents initial data collection and 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. We then present semantic labeling experiments using a semi-Markov conditional random field (CRF) that obtains an F1 test score of 85.1. Finally, we report food segmentation experiments that explored three methods for associating foods with their corresponding attributes: a generative Markov model, transformation-based learning, and a CRF classifier. The CRF performed best, achieving an F1 test score of 87.1.

Index Terms— Data collection, Semantic tagging, CRF, Markov model, Transformation-based learning

1. INTRODUCTION

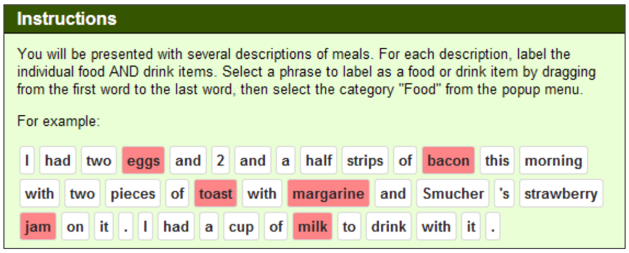
Existing approaches for the prevention and treatment of obesity are hampered by the lack of accurate, low-burden methods for self-assessment of food intake, especially for hard-to-reach, low-literate populations [1, 2]. For this reason, we have begun to explore whether speech understanding and dialogue technology can enable efficient self-assessment of energy and nutrient intake. We are interested in studying whether speech can lower user burden compared to existing self-assessment methods, whether spoken language descriptions of food intake can accurately quantify caloric and nutrient intake, and whether dialogue can efficiently and effectively be used to ascertain and clarify food properties, perhaps in conjunction with other modalities.

In this paper, we describe our current progress in the extraction of food concepts from a user's spoken meal description (e.g., extracting "a bowl of Kellogg's cereal" from the food log "This morning for breakfast I had a bowl of Kellogg's cereal"). Specifically, we focus on the crowdsourced data collection and annotation we have performed in order to create an initial repository of semantically annotated food logs. We then describe the experiments we have conducted for the tasks of semantic labeling and segmentation on these data. The understanding component forms part of a larger nutrition logging prototype 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, description, etc.) associated with particular word sequences. The segmented food concepts are then shown in matrix form in a table along with potential matches to a nutritional database containing over 20,000 foods from the USDA and other sources.

In the remainder of this paper, we begin by describing the crowdsourcing methods we have developed and deployed on Amazon Mechanical Turk (AMT) for data collection and annotation [3]. Section 3 provides details on the language understanding techniques we have explored, and Section 4 reports experimental results. Finally, Section 5 concludes and describes our future plans.

2. DATA COLLECTION

We deployed three phases of experiments on AMT in order to crowdsource our data collection and annotation. The first phase involved the collection of food diaries, where we prompted Turkers to write a description of a meal as they would imagine describing it orally. The diaries were then tokenized and used as input for the second phase, shown in Figure 1, where we asked users to label individual food items within the diaries. The third phase combined the food diaries with their food labels and prompted Turkers to label the concepts associated with a particular food item (see Figure 2).



Instructions

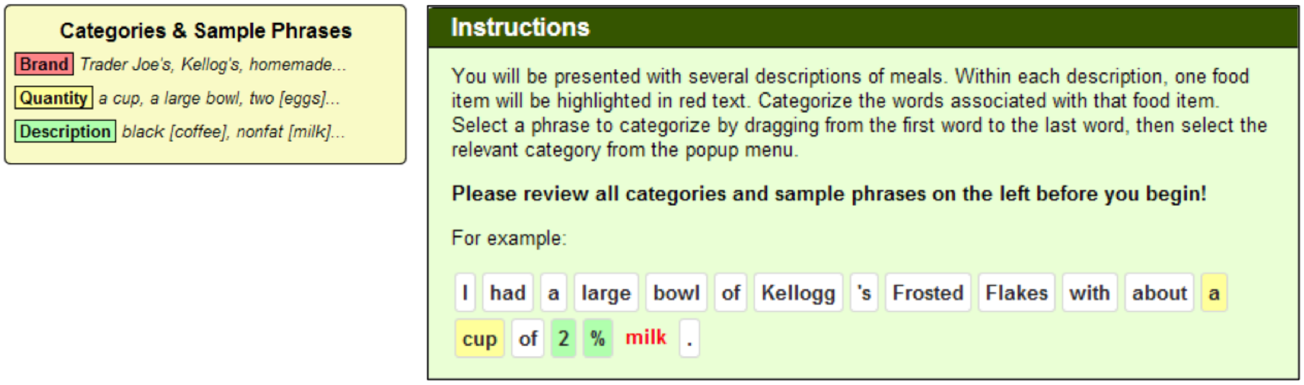
You will be presented with several descriptions of meals. For each description, label the individual food AND drink items. Select a phrase to label as a food or drink item by dragging from the first word to the last word, then select the category "Food" from the popup menu.

For example:

I had two eggs and 2 and a half strips of bacon this morning with two pieces of toast with margarine and Smucher 's strawberry jam on it . I had a cup of milk to drink with it .

Fig. 1. The AMT task for labeling foods in a meal description.

After the initial round of data collection, we noted that Turkers were producing food diaries of lower quality than we desired. In order to improve the descriptions, we required



Categories & Sample Phrases

Brand	Trader Joe's, Kellog's, homemade...
Quantity	a cup, a large bowl, two [eggs]...
Description	black [coffee], nonfat [milk]...

Instructions

You will be presented with several descriptions of meals. Within each description, one food item will be highlighted in red text. Categorize the words associated with that food item. Select a phrase to categorize by dragging from the first word to the last word, then select the relevant category from the popup menu.

Please review all categories and sample phrases on the left before you begin!

For example:

I had a large bowl of Kellogg 's Frosted Flakes with about a cup of 2 % milk .

Fig. 2. The AMT task for labeling properties of foods.

the food diaries to pass a series of checks before submission. Our algorithms address several common trends we identified among low-quality diaries.

Often, a single entry was submitted numerous times, resulting in semantically identical data. Our solution was to generate a corpus of submitted responses and disallow repeat submissions. In addition, low-quality descriptions often contained few words, so we required diaries to consist of at least four words. Another attempt to outwit the checker involved using repetition within a diary (e.g., “a a a a”). Our solution to this challenge was to prevent diaries from containing more than 60% repetition. Finally, due to extensive spelling errors, for submission we required at least 60% of the words in the description to match entries in an English dictionary.

We collected and labeled 1,302 breakfast diaries, which we used to train our models. The data were tokenized by the European Parliament Proceedings Parallel Corpus tokenizer [4]. We used the AMT label for a token if at least a threshold of four out of five Turkers labeled the token as a food item or if three out of five Turkers labeled the token as the same attribute. The thresholds were selected by comparing the performance of the resulting model trained on these labels, as shown in Table 1. Every tenth query was added to the test set (for a total of 131 test queries), while all other queries are part of the training data (1,173 training queries in total). The histogram in Figure 3 shows that most food diaries contain three, four, or five foods. Turkers tend to have high agreement when labeling foods and quantities, but there are more conflicts among brands and descriptions.

To ensure consistency between the AMT data and the recognized food diaries in the deployed nutrition system, we created a data pre-processing step to normalize the tokenization and punctuation. Since the recognizer results include apostrophes and percent signs, but no commas or periods, we removed all punctuation except apostrophes and percents from the labeled data. In addition, the text displayed in the nutrition system does not split punctuation into separate tokens, so we combined punctuation tokens with the previous word tokens.

Threshold	Mean F1	Variance	St. Dev.
1	78.75	2.21	1.49
2	84.05	2.38	1.54
3	**84.58**	2.81	1.68
4	83.74	1.01	1.01
5	76.80	2.78	1.67

Table 1. Labeling model’s 10-fold results (i.e., mean F1, variance, and standard deviation) for different thresholds of Turkers in the property labeling task. The threshold of three Turkers achieves the highest mean F1 score (shown in bold).

3. METHODS AND EXPERIMENTS

The language understanding component of the nutrition system has two phases: semantically labeling the food concepts and properties in a meal description, and assigning attributes to the correct food items. Previous work has applied frame-semantic parsers [5], triangular CRFs [6], and neural networks [7, 8] to the problem of semantic tagging.

3.1. Labeling

To accomplish the first semantic tagging task, we utilized a variation of the standard CRF model, a semi-Markov conditional random field (semi-CRF). Rather than assigning an output label to each token, a semi-CRF assigns an output label to token *segments* [9].

Semi-CRFs can be viewed as the conditional, or discriminative, version of generative semi-Markov chain models, in which there is a segment of tokens from i to d_i where the behavior of the system may not be Markovian. The state s_i at token i persists until token d_i, at which point there is a transition to a new state s' which only depends on state s_i. Semi-CRFs perform better on segmenting tasks such as named entity recognition and noun-phrase (NP) chunking. In our case, the semi-CRF is a reasonable choice for a semantic tagging

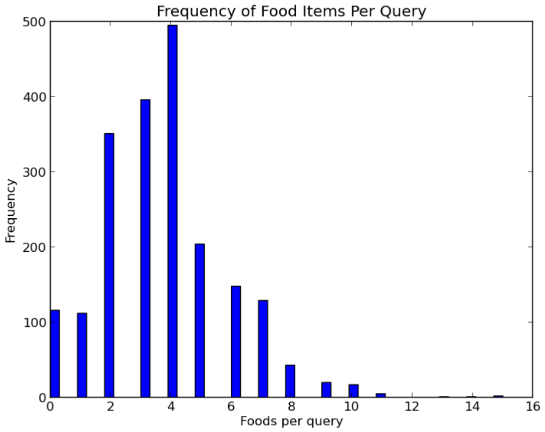


Fig. 3. Frequency of food items per meal description.

model because food items and properties are sequences of tokens. For example, a quantity might be the segment "a cup."

Rather than modeling the conditional probability of output $\mathbf{y}$ given input $\mathbf{x}$, $Pr(\mathbf{y}|\mathbf{x})$, a semi-CRF models the probability of a segmentation $\mathbf{s}$ given $\mathbf{x}$, $Pr(\mathbf{s}|\mathbf{x})$, where each segment $s_i \in \mathbf{s}$ consists of a start position t_j, an end position u_j, and a label y_j: $s_i =< t_j, u_j, y_j >$. For example, the quantity segment "a cup," appearing in the food log "I had a cup of milk," would be represented as $< 2, 3, Quantity >$, assuming zero-indexing, since "a" has index 2 and "cup" has index 3. Also, rather than using local feature functions $\mathbf{f}$, which correspond to output labels of individual elements in $\mathbf{x}$, semi-CRFs use segment feature functions which correspond to output segments of $\mathbf{x}$. We define each segment feature function $g^k(j, \mathbf{x}, \mathbf{s}) = g^k(y_j, y_{j-1}, \mathbf{x}, t_j, u_j)$ according to the Markov assumption that a segment s_j depends only on the previous segment s_{j-1}. Then, if we let $\mathbf{G}(\mathbf{x}, \mathbf{s}) = \sum_{j=1}^{|\mathbf{s}|} \mathbf{g}(j, \mathbf{x}, \mathbf{s})$, a semi-CRF estimates the distribution

$$Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}) = \frac{1}{Z(\mathbf{x})} \exp\left(\mathbf{W} \cdot \mathbf{G}(\mathbf{x}, \mathbf{s})\right) \qquad (1)$$

where $\mathbf{W}$ is a weight vector for $\mathbf{G}$ and $Z(\mathbf{x})$ is the normalization factor $\sum_{\mathbf{s}'} \exp\left(\mathbf{W} \cdot \mathbf{G}(\mathbf{x}, \mathbf{s}')\right)$. In addition, semi-CRFs provide the benefit of high-order CRFs without the associated computational cost.

The features selected for the food/property labeling task include n-grams (unigrams, bigrams, trigrams, and 4-grams), lexical features (i.e., the segment matches an item in a lexicon of USDA food products), and part-of-speech (POS) tags. We used Stanford's open source tagger to generate POS tags [10].

3.2. Segmenting

In the second phase of the language understanding component, we took the semi-CRF output and determined which attributes were associated with which foods. We investigated three approaches: a Markov model (MM), transformation-based learning, and a CRF classifier.

3.2.1. *Simple Rule*

As our baseline, since 90.8% of the attributes in the data appear prior to their corresponding food item, we defined a simple rule which assigns properties to the subsequent food. For example, in the description "I had a cup of milk with a handful of blueberries," the quantity "a cup" would be associated with "milk," and the quantity "a handful" with "blueberries." In the case where an attribute appears after the last food item in the description, the attribute is assigned to the last food.

3.2.2. *Markov Model*

To improve upon the baseline method, we took advantage of the sequential nature of the food description data (e.g., a food item may be more likely to appear after a brand than a quantity) by modeling it probabilistically. We defined a first-order Markov chain, in which each observation x_i depends only on the previous observation x_{i-1}. The joint distribution for a sequence of n observations under this generative model is

$$p(x_1, ..., x_n) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}), \qquad (2)$$

which leads to the conditional distribution for observation x_i, given all previous observations, of

$$p(x_i|x_1, ..., x_{i-1}) = p(x_i|x_{i-1}), \qquad (3)$$

since, by the Markov assumption, x_i depends only on the previous observation x_{i-1} [11, 12]. In our case, we let each observation in the Markov chain represent an attribute or food item. For example, in the meal description "I had a bowl of cereal," the semi-CRF would label "a bowl" as a quantity and "cereal" as a food, resulting in the Markov chain in Figure 4.

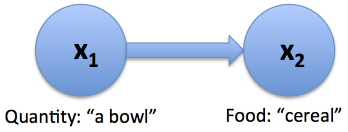


Fig. 4. A first-order Markov chain for the food description "I had a bowl of cereal."

We implemented this Markov model with a finite state transducer (FST), which transduces an input string into an output string [13]. Like a Markov model, an FST has states and transitions with associated weights; we let each state represent a possible food/property semi-CRF label and added start and end states. The input to the FST is a string of food/property labels in a food diary, and the output is the same string segmented by "#" such that each food item and its associated properties are within the same segment. For example, the diary "I had a bowl of cereal with milk" would correspond to the input string "Q F F" and would generate the output "Q F # F." This indicates that "a bowl" is a quantity describing "cereal," and "milk" has no attributes.

We used the frequency of label patterns (e.g., "Q B F D F") that occur in the training data to calculate the initial state distribution (e.g., $P(Q) = 0.65$) and the state transition probabilities (e.g., $P(F|D) = 0.57$). In addition, using the histogram in Figure 3, we calculated the probability that a subsequent food follows the current food. The transition weights in the FST behave like negative log probabilities.

3.2.3. Transformation-Based Learning

We further improved upon both the simple rule and the Markov model approaches by applying a transformation-based learning (TBL) algorithm. This method starts with an initial solution to a problem (e.g., the simple rule baseline) and iteratively applies transformations, selecting those which improve the performance most. We used the Fast TBL toolkit developed at Johns Hopkins [14].

In order to adapt TBL to the food-property association problem, we framed it as a classification task. To do this, we modeled it after the NP chunking problem, a well-known natural language processing (NLP) task. In NP chunking, each word in a sentence belongs to one of three classes: B (the start of 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 data samples from which the classifier learns are composed of a token, its semi-CRF label (i.e., food, quantity, brand, description, or other), the predicted chunk label, and the actual chunk label. An example food diary is shown in Table 2. We also defined general rule templates from which the model learns specific rules that may be applied in order to improve the system's performance. For example, the rule template "chunk_0 chunk_1 $\text{label}_0 \Rightarrow$ chunk" implies that given the current chunk label, the next chunk label, and the current semi-CRF label, the current chunk label should be transformed to that specified by the rule.

Token	CRF Label	Chunk
I	Other	O
had	Other	O
a	Quantity	B
bowl	Quantity	I
of	Other	I
cereal	Food	I

Table 2. Example of the food chunking classification problem, where a chunk label B, I, or O is assigned to each token, given its semi-CRF label (i.e., brand, quantity, description, food, or other).

3.2.4. CRF

As an alternative to the TBL algorithm, we investigated the CRF. Since it is a discriminative classifier as well, the CRF was trained on the same data as the TBL model. However, rather than defining a set of rule templates, we provided feature templates corresponding to unigrams and bigrams of output tags that appear with certain combinations of tokens and food/property labels. We used the CRF++ toolkit [15] and also trained a TBL model with the CRF as a baseline.

4. RESULTS

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, recall, and F1 (harmonic mean of precision and recall) scores for each approach.

4.1. Labeling

Through 10-fold cross-validation, we selected the set of features for the semi-CRF that resulted in the highest average F1 score. As shown in Table 3, the F1 scores of various feature sets are all similar. We selected n-grams (i.e., bigrams, trigrams, and 4-grams), food lexicon features, and POS tags.

Features	Mean F1	Var	St. Dev.
N-grams	84.57	0.57	0.75
+ Food lexicon	84.62	1.25	1.12
+ POS tags	**84.83**	1.17	1.08

Table 3. Semi-CRF 10-fold results for different feature sets. N-gram, lexicon, and POS tag features combined achieve the highest mean F1 score (shown in bold).

We measured the performance of the semi-CRF trained on our selected feature set using the test data, as shown in Table 4. We evaluated the semi-CRF at the concept level as opposed to the word level so that a concept is considered correct if the CRF labels the concept correctly, even if a word within the concept is labeled incorrectly (i.e., the current token's label is the same as either the previous or the next token's label, and both the previous and next tokens' labels are correct). For example, "a bowl" would be counted as correct even if "a" is labeled incorrectly, as long as "bowl" is labeled correctly.

Examples of the types of errors made by the semi-CRF are shown in Table 5. Descriptions and brands are sometimes swapped or omitted altogether. From Tables 4 and 5, we infer that the semi-CRF identifies foods, quantities, and other much more easily than brands or descriptions, which reflects the high Turker disagreement for the brand and description

Label	Precision	Recall	F1
Food	92.45	87.50	89.91
Brand	87.32	70.99	78.32
Quantity	92.43	91.33	91.87
Description	85.57	77.57	81.37
Other	91.66	95.76	93.67
Overall	88.25	82.16	85.09

Table 4. Semi-CRF concept-level performance on test data.

categories. In addition, some brands may not be seen in training data (i.e., out-of-vocabulary words). To address these issues, we may need to revise the AMT tasks to enable Turkers to more easily differentiate between brands and descriptions. We can also create a brand lexicon using the nutritional database. In the future, the nutrition system may learn new brands or foods through dialogue with a user.

Token	Predicted Label	AMT Label
Nescafe	Description	Brand
frosted	Brand	Description
Ritz	Other	Brand

Table 5. Semi-CRF semantic tagging errors.

We compared the semi-CRF performance to that of a standard CRF baseline (using the CRF++ toolkit), testing for significance at the $p < 0.0001$ level using McNemar's significance test [16]. We found that the difference was not statistically significant; thus, although the semi-CRF handles segmenting in a more intuitive manner than do standard CRFs, there is no significant gain.

4.2. Segmenting

In Table 6, we present the performance of six approaches to the food-property association task. The TBL algorithm significantly improved upon the simple rule and Markov model (MM) baselines, but not the CRF (where statistical significance was measured using McNemar's test). Although the Markov model performed slightly better than the simple rule, the difference is not statistically significant, whereas the improvement of the CRF over all other methods is statistically significant ($p < 0.0001$). The CRF model performed best, achieving a token-level accuracy of 97.22% and a phrase-level F1 score of 87.13.

A few examples of errors made by the different models are shown in Table 7. Since brands and descriptions occasionally appear after a food, the simple rule incorrectly assigns attributes in these cases. For example, in the food diary "two eggs sunny side up," the simple rule mistakenly begins a new food chunk with the description "sunny side up," since "sunny side up" is actually part of the food chunk corresponding to "eggs." Although TBL corrects many of these errors, some still remain (e.g., it predicts two food chunks for "grits Quaker" instead of one). MM errors occur when it incorrectly segments the labels. For example, it segments the food diary "oatmeal from Dunkin' Donuts I had water to drink" into the output string "F # B F," which incorrectly associates the brand "Dunkin' Donuts" with "water" rather than "oatmeal."

Approach	Acc	Prec	Recall	F1
Simple Rule	84.44	51.50	54.22	52.83
Simple + TBL	94.31	77.94	78.27	78.11
MM	84.86	54.64	57.17	55.88
MM + TBL	95.22	82.65	80.38	81.50
CRF	**97.22**	**87.13**	**87.13**	**87.13**
CRF + TBL	95.48	83.97	82.91	83.44

Table 6. Test performance of approaches to the food segmenting task, where accuracy is calculated at the token-level and precision, recall, and F1 are computed at the phrase-level. The CRF (shown in bold) achieved the best performance.

Method	Text	Auto	AMT
Simple	two eggs sunny side up	BIBII	BIIII
+ TBL	grits Quaker	BB	BI
MM	oatmeal from Dunkin' Donuts	BOBI	BIII
+ TBL	cake that i bought	IIII	IOOO
CRF	butter to grease	BII	BOO

Table 7. BIO food chunking mistakes, where auto is the prediction and AMT is the gold standard annotation.

The TBL toolkit outputs a list of successful rules, as shown in Table 8. This allows us to observe where the baseline method made errors and which rules were used to fix those mistakes. For example, the model learned that if the previous two chunk labels are O, then the current chunk should change from I to B, since new food chunks always start with B. In addition, if the current token is labeled Other but has the chunk label B, then the chunk label should be changed to I because a new food chunk cannot start with the label Other. Some rules specify that attribute tokens should have the chunk label I rather than O, which is reasonable since attributes should be part of food chunks.

Rule	Score
$C_{-2} = O \, C_{-1} = O \, C_0 = I \Rightarrow C = B$	351
$C_0 = B \, L_0 = O \Rightarrow C = I$	172
$C_0 = O \, L_0 = D \Rightarrow C = I$	169

Table 8. TBL (with simple rule baseline) high-scoring rules, where score is the number of improvements minus performance reductions. C_i represents a chunk label at index i (e.g., B, I, or O), and L_i indicates the food/property label at i.

5. CONCLUSIONS

We have described experiments for extracting food concepts from spoken input to a nutrition dialogue system. We explained the process of data collection via AMT crowdsourcing and presented two phases of the system's language understanding component: semantic tagging and association of attributes with foods. We measured the performance of a semi-CRF that outputs segments of foods and attributes, and described the selected features. We also evaluated three approaches for assigning properties to foods: a Markov model that segments food concepts with their attributes, a TBL algorithm that iteratively learns rules to correct the baseline's errors, and a CRF classifier that frames the food segmentation task as a chunking problem.

Since the majority of attributes appear prior to their corresponding food items, the baseline simple rule which associates foods with prior attributes performs similarly to the Markov model approach. However, the simple rule makes mistakes which the Markov model does not. For example, brands and descriptions may appear after a food; in the diary "I had eggs from Trader Joe's with bread," "eggs" is the food item and "Trader Joe's" is its brand, but the simple rule would assign "Trader Joe's" to the food "bread." Even though the Markov model is likely to segment these diaries correctly, the TBL algorithm shows greater improvement by directly correcting these errors through the use of transformation rules.

Transformation-based learning also improves upon the Markov model by learning rules that fix typical errors made by this model. TBL and the CRF contain more information than the Markov model by incorporating tokens, not just the food/property labels, as well as tokens that are labeled Other. The CRF is the best model, labeling the test data with the highest F1 score of 87.13, possibly because the food chunking problem mirrors the standard NP chunking problem.

In the future, we plan to focus on other components of the nutrition system in addition to language understanding. Specifically, a dialogue manager may interact with the user by asking follow-up clarification questions in order to select the top food from the list of 10 hits returned by the nutritional database or to learn new foods and brands when it encounters out-of-vocabulary words. In addition, we will need to map the user-described quantities (e.g., "smothered in") to the quantities listed in the database (e.g., 1 tablespoon) so that we can extract the correct nutrition facts. Finally, we plan to incorporate a bar code scanning feature, refine the user interface, and potentially use computer vision to help detect food quantities.

6. ACKNOWLEDGEMENTS

Jeffrey Sun, Daniel Li, and Panupong Pasupat helped with the development of early versions of the data collection tasks for AMT. This research was sponsored in part by a grant from Quanta Computing, Inc., and by the NIH.

7. REFERENCES

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

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

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

[4] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, 2005, vol. 5, pp. 79–86.

[5] Y. Chen, W. Wang, and A. Rudnicky, "Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing," in *Proc. ASRU*, 2013, pp. 120–125.

[6] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *Proc. ASRU*, 2013, pp. 78–83.

[7] 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.

[8] A. Deoras and R. Sarikaya, "Deep belief network based semantic taggers for spoken language understanding," in *Proc. INTERSPEECH*, 2013, pp. 2713–2717.

[9] S. Sarawagi and W. Cohen, "Semi-markov conditional random fields for information extraction," in *Proc. NIPS*, 2004, pp. 1185–1192.

[10] M. De Marneffe, B. MacCartney, C. Manning, et al., "Generating typed dependency parses from phrase structure parses," in *Proc. LREC*, 2006, vol. 6, pp. 449–454.

[11] C. Bishop et al., *Pattern Recognition and Machine Learning*, vol. 1, Springer New York, 2006.

[12] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," *Introduction to statistical relational learning*, pp. 93–128, 2006.

[13] I. Hetherington, "The MIT finite-state transducer toolkit for speech and language processing," in *Proc. INTERSPEECH*, 2004.

[14] R. Florian and G. Ngai, "Fast transformation-based learning toolkit," *Johns Hopkins University, USA*, 2001.

[15] T. Kudo, "CRF++: Yet another CRF toolkit," *Software available at http://crfpp. sourceforge. net*, 2005.

[16] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, 1989, pp. 532–535.