Ingredient, Unit and Amount Named Entity Recognition and Relationship Identification in Cooking Speech Transcripts

X. Chu, Y. Xia, J. Yang, D. Zhang, X. Zhang

Abstract: In this project, the meaningful entities and their relations in cooking video transcripts were tagged. A CRF model was trained with the tags, to predict the entities and relations in both non-automatic and automatic transcripts. The model achieved satisfactory F-measure on non-automatic transcripts and high Precision on automatic transcripts. However, the Recall obtained in automatic transcripts was expectedly not that high. The analysis of the model performance on ingredient entity recognition and relationship identification was further developed.

1. Goal

The purpose of our project is to build a system that could automatically recognize meaningful entities in cooking tutorial videos (including recipe names, ingredients and amounts and units), as well as the relationship between quantities, units and ingredients. In the future the trained model can be adopted to build applications to automatically generate recipe summary from videos, and hence to support decision making of rookie home chefs.

2. Corpora

2.1 Meta information

The corpora were extracted from YouTube using 5 search queries: Laura in the Kitchen, Healthy Recipe Videos, Hilah Cooking, Eugenie Kitchen and Stephanie Manley.

2.2 Statistics

In this section, the detailed statistic of chosen transcripts is suggested as follows:

	non-automatic	automatic
Transcript count	33	33
Token count	53989	43445[1]
Type count	3157	4695
Normalized type count	2508	3815
Sentence number	3772	N.A.[2]

Note: [1] Token count: punctuation marks are counted as token. [2] Automatic sentence number cannot be determined since no punctuation is provided in automatic transcript according to this. Current recognition systems still output merely a stream of words. The unannotated word stream lacks useful information about punctuation. However, in later section, the CRF model developed can still perform entity recognition with great accuracy without punctuations and sentence segmentations.

3. Training Data Preparation

3.1 Methodology

3.1.1 Techniques & Tools

Tags for entities and relations were manually added on the non-automatic transcript with the *BRAT rapid annotation tool*. The tool is set up on a cloud server (http://app.deepreader.io:8001/) for the team to access and conduct the annotation collaboratively.

3.1.2 Specifications

3.1.2.1 Definition of entities

Recipe: The particular recipe name for the video.

Ingredient: The name of raw materials which the recipe is made from. Mixtures or transformations of sets of ingredients (such as 'dough') do not fall into this category.

Amount: The quantity of ingredients used. Both specific amount like 'two', '4 to 5'and general amount like 'some' are included.

Unit: The unit measure of ingredients, both standard measures like 'cup' and non-standard measures such as 'pinch' and 'bit'.

3.1.2.2 Definition of binary relations

Quantify**:** Arg1: Amount, Arg2: Ingredient

Measure**:** Arg1: Unit, Arg2: Ingredient

3.2 Inter-Annotator Agreement (IAA)

3.2.1 Code for Kappa value calculation

	I2	I3	I4
Kappa Value	0.7726	0.8514	0.8887

3.2.2 Observation from Changes from I2 to I4

From the table above, it is drawn that the kappa value has been improved from 0.7726 to 0.8514, and finally to 0.8887. The significant increment of 0.08 between I2 and I3 is mainly produced due to the consensus reached among two sub-teams on the exact definitions of ‘Amount’ and ‘Units’ Entities after discussing the differences in S2’s annotations. For instance, words representing general amount like 'some' and ‘a little' were confirmed to be labeled as ‘Amount’ entity. The definition of ‘Unit’ was also refined to include

non-standard units such as 'bit' and 'pinch'. The improvement from I3 to I4 shows a relatively smaller value of 0.03. This slight improvement is due to some peripheral agreements made after discussing S3's annotation, for instance, labeling general term 'veggies' and 'meat' as Ingredients.

4. NER Model Development

4.1 Methodology

Stanford NER is a CRF Classifier, which provides a general implementation of (arbitrary order) linear chain Conditional Random Field sequence models. Stanford NER does not require manual provision of f POS tag and Syntactic tag, thus POS tag and Syntactic tag are not generated.

Stanford NER is used for:

- Model training using training data (automatic transcripts), with 4-fold cross-validation.
- Model prediction for non-automatic transcripts.

4.2 Model Feature Inclusions & Exclusions

4.2.1 Disabling Word Shape feature

The entities identified do not rely on the specific word shapes. For example, the ingredient entities do not need word shape features like capitalization as needed in normal NER for location or people name recognition. Consequently, the word shape feature in CRF model is turned off.

4.2.2 Lemma & Word Normalization

Normalization is set to be true for the CRF model. Thus, some tokens are normalized when processed by the CRF model. At the same time, the lemma of the word is provided as a feature for CRF. Word lemma as a feature and normalization are desirable since ingredients, unit and recipe are normal English words. It is more suitable since "egg" should not be discriminated from "eggs".

4.2.3 CoNLL IOB2 format

The entity sub-classification is enabled and specifically the model is using IOB2 format. The labels of entity type are converted into IOB2 encoding with a B*(Beginning)* and I*(Inside)* sub-classification for each entity type. In a sequence model, for IOB2, the labels are treated as different classes.

For raw data, the manual tagging of "B-", "I-" is annotated by the human observers with BRAT tool. Subsequently, the resulting tag set for the entity type is used for training and classification in the model.

4.2.4 Dictionary as a Feature for CRF

Most of cooking ingredients can be enumerated; thus a dictionary for Stanford NER is provided describing the common ingredients according to food list. Specifically, Stanford NER use term gazette for the dictionary feature. Thus the NER model here is trained with gazette features. Gazette is supplied at training time; hence the NER model will learn features based on words or phrases provided. The gazette used is here.

By using gazette, the model's F-measure increases with around 3~4% for ingredient tag. See details here.

4.3 Cross Validation Results for Non-auto Script

Fold 1:

Entity	P	R	F1	TP	FP	FN
Amount	0.7838	0.7073	0.7436	58	16	24
Ingredient	0.7729	0.6062	0.6795	177	52	115
Recipe	0.0000	0.0000	0.0000	0	1	46
Unit	0.7955	0.8333	0.8140	35	9	7
Totals	0.7759	0.5844	0.6667	270	78	192

Fold 2:

Entity	P	R	F1	TP	FP	FN
Amount	0.8039	0.5775	0.6721	82	20	60
Ingredient	0.7179	0.6275	0.6697	224	88	133
Recipe	0.0000	0.0000	0.0000	0	2	26
Unit	0.7778	0.6447	0.7050	49	14	27
Totals	0.7411	0.5907	0.6574	355	124	246

Fold 3:

Entity	P	R	F1	TP	FP	FN
Amount	0.7967	0.6323	0.7050	98	25	57
Ingredient	0.7883	0.5307	0.6344	216	58	191
Recipe	0.0000	0.0000	0.0000	0	1	23
Unit	0.9184	0.7627	0.8333	90	8	28
Totals	0.8145	0.5747	0.6739	404	92	299

Fold 4:

Entity	P	R	F1	TP	FP	FN
Amount	0.8846	0.5349	0.6667	69	9	60
Ingredient	0.8469	0.5268	0.6495	177	32	159
Recipe	0.0000	0.0000	0.0000	0	0	0
Unit	0.9245	0.7903	0.8522	49	4	13
Totals	0.8676	0.5148	0.6462	295	45	278

Details of result can be found here.

4.4 Discussions

The project model achieved an overall 64%-67% level of accuracy in non-automatic script name entity recognition. The detailed breakdown is as follows: 1) the recognition of unit entity achieved a high accuracy of 70%-85%; 2) the recognition of amount entity achieved 66%-74% level of accuracy; 3) the recognition of ingredient entity achieved 63%-68% level of accuracy. The low accuracy of amount recognition is largely affected by the tagging rule where only

ingredient-relative amounts were tagged. Since the NER cannot distinguish whether an amount is ingredient-relative or ingredient-irrelative while being trained, the accuracy of the amount recognition diminished. The potential solution is to tag all amount regardless of the relationship with ingredients while additional programs need to be developed to identify the relationship with ingredients. The focus of this project is on ingredient entity recognition, which occupies the highest number of tokens in the transcripts, and the various factors affecting the accuracy of ingredient are discussed as follows:

4.4.1 Mixture and Intermediate Blend

The mixture/blend refers to any food within the recipe that are mixtures or bi-products made from ingredients or even spices (i.e. ‘dough', 'sauce'). Most mixtures and blends are ignored by the NER tagger. However, there are 7 instances out of all scripts that the mixture or blend is marked as ‘Ingredient’, i.e. ‘vanilla dough’ and ‘tartar sauce’. This type of prediction increase FP for ingredient tag. A straightforward solution is to introduce a new tag ‘mixture’ which specifically trains the module to identify the mixture/blend entities. The new tag is expected to be effective in distinguishing entities in the form of ‘mixture’ or ‘blend’ (reference). However, distinguishing between ingredients and mixtures is even ambiguous for human. It is difficult to differentiate whether the ‘tartar sauce’ is homemade or store bought without diving deep into the context. Relevant effort of adding the ‘mixture’ tag will be disseminated to future research groups.

4.4.2 Entity Description

In some scripts, there are many words that modify the type of food mentioned, such as 'chopped' before the ingredient 'beef'. In this project, this type of general description is not tagged *(neither included in the Ingredient tag nor an additional description tag)* since the focus of the project is to help users cook with store bought ingredients. As far as store-bought ingredient can be processed to the described form, the users will still be able to cook the particular dish. For instance, among ‘cold water', where 'water' is tagged as ingredient while 'cold' is not.

On the other hand, the exclusion of some descriptions may result in using totally different ingredients. In the sentence 'so you all do use bread flour', merely tagging 'flour' is not meaningful enough since the consumers will accidently use all-purpose flour. In this case, 'bread' is included in this ingredient as ‘bread flour’. A more complex example is 'natural peanut butter', where 'natural' can be excluded and 'peanut butter' is tagged.

Adopting this complex tagging rule for ingredient entity acquires only meaningful information of the store bought ingredients. Nevertheless, this tagging methodology surges the complexity of NER tagging. NER model cannot effectively distinguish between the description and ingredient. For example, in “white onion”, only “onion” is tagged while the “white” is not, which is the dominant cause of high ingredient entity FN.

In total, our model correctly predicts the ingredient in the accuracy level of 63.44% - 67.95% (***F1)***. Further studies are encouraged to separately tag “description” and classify description into “Meaningful” and “Non-meaningful”.

4.4.3 Recipe Identification

Recipe tag is the least frequently occurred tag in our project scripts. Only exact match and abbreviation of the recipe name is tagged in manually tagged scripts. The poor accuracy of recipe recognition is mainly due to small amount of training. Consequently, it is more appropriate to identify Recipe in the meta data (e.g. titles) of the video rather than in the transcript itself.

4.4.4 Context Dependent Entity Recognition

Sometimes, there are ingredients mentioned as alternatives or negative examples. Alternatives or negative examples include “not, better than, like, instead, or” etc. When this happens, the manual tagging follows the semantics of the sentence; all suggested alternatives were tagged as ingredients, while negative examples were omitted. As a consequence, NER model has difficulty differentiating normal items and negative examples, which increases the FP.

4.4 Prediction Results for Automatic Script

The automatic scripts were also fed to the CRF library and the result was checked against the tags done manually on the non-automatic scripts. Some difficulties emerged during the process as the automatic transcripts were of low quality: some sentences and words were missing or recorded wrongly. Even if a word was correctly tagged in both non-automatic and automatic scripts, their actual position may be completely different. Hence, the 1-to-1 exact comparison of the tags became highly time-consuming and the result was untenable.

The solution proposed for this problem was to use a simplified comparison scheme: only the entity tags and

their numbers are compared. FP cases are defined as: 1) entities that only appeared in the automatic transcripts or 2) if the number of the same entity tag is larger in automatic transcript than the non-automatic, the difference in the numbers. TP and FN cases are defined in the same way, and the measurements were calculated with this definition. It can be observed that Amount, Unit and Ingredient entity can be recognized with satisfiable precision. However, the recall is low. This may be caused by the misspelling in the automatic transcripts. Another problem is that the Recipe is not picked up at all. The reason is that there is extremely low amount of recipe tags in the training set, as discussed in the previous section.

	Precision	Recall	F-measure
Amount	0.8148	0.2165	0.3421
Recipe	0.7500	0.1915	0.3051
Unit	0.9851	0.2193	0.3587
Ingredient	0.8641	0.3700	0.5181

4.4.1 CRF Prediction for Automatic Transcripts

Prediction on the automatic transcripts was done in a 4-fold manner. For instance, when the model is trained with A1, A2, and A3, then the test is performed on A4. A full round of training and testing is performed for all automatic transcripts.

The 4-fold manner was to ensure that the non-automatic version of the scripts in the test set do not appear in the training set, as the non-automatic and automatic versions are essentially the same with some degrees of variance. Otherwise, the NER model trained with the entire non-automatic transcripts can achieve high F-measure since essentially the model predicts the same data as the training data although some variations. If the NER model is trained with the entire training data, the result is shown as followed:

	Precision	Recall	F-measure
Amount	0.7684	0.1437	0.2421
Recipe	0.0000	0.0000	0.0000
Unit	0.9643	0.1794	0.3025
Ingredient	0.7357	0.2680	0.3928

5. Entity Relationship Model Development

5.1 Methodology

After NER system tagged the entity name on the scripts, the relation between the entities could be extracted. For this recipe case, the amount of entities could be extracted from the tagged script.

Regular expression is used to extract the amount relation. There are several rules to find a relation:

```
<amount><ingredient>
<amount>[some description words without tag ]<ingredient>
<amount><unit><ingredient>
<amount>[some description words without tag ]<unit><ingredient>
<amount><unit>[some description words without tag ]<ingredient>
<amount>[some description words without tag ]<unit>[some description words without tag ]<ingredient>
<amount>[some other ingredient]<ingredient>
<amount><unit>[some other ingredient ]<ingredient>
```

The above rules could be summarized with following regular expressions:

```
<Amount>(?P<amount>[^<]+)</Amount>[^<]{0,70}(<Ingredient>[^<]+</Ingredient>)*<Ingredient>(?P<ingredient>[^<]+)</Ingredient>
<Amount>(?P<amount>[^<]+)</Amount>[^<]{0,70}<Unit>(?P<unit>[^<]+)</Unit>[^<]{0,70}(<Ingredient>[^<]+</Ingredient>)*<Ingredient>(?P<ingredient>[^<]+)</Ingredient>
```

The full code is attached in info_extraction.py .

5.2 Results

For each test file, the extracted relations is compared with the relations in the manually tags.

The system is tested with both manually tagged scripts and NER tagged scripts. The result is shown in table below:

Data Set	**TP**	**FP**	**FN**	**F1**	**Recall**
Manual	365	47	49	0.883777	0.881643
NER Model	211	71	203	0.606322	0.509662

5.3 Discussion

The result with manually tagged scripts shows the accuracy of the relation-extraction model. The overall accuracy amounts to 88%, which performs well for an information extraction tool. There are several limits for yield even higher accuracy. 1) The uses of pronoun. 2) Multiple amount entities and unit entities related to the same ingredient entity. 3) The long distance between a ingredient entity and its amount and unit entities. 4) Some noises or coincidence.

When the tool is tested with NER tagged scripts, the accuracy drop significantly, the main reason is that there is high chance for the NER model to miss some entities which should be tagged during the tagging phase. And to extract a correct relation, it is required to tag the amount, unit and ingredient entities correctly at the same time. As a result, this tool imposes a high likelihood to miss a relation and obtain a FN result.