Recipe Instruction Semantics Corpus (RISeC): Resolving Semantic Structure and Zero Anaphora in Recipes

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, Chris Develder
Ghent University – imec, IDLab
Ghent, Belgium
{first_name.last_name}@ugent.be

Abstract

We propose a newly annotated dataset for information extraction on recipes. Unlike previous approaches to machine comprehension of procedural texts, we avoid a priori pre-defining domain-specific predicates to recognize (e.g., the primitive instructions in MILK) and focus on basic understanding of the expressed semantics rather than directly reduce them to a simplified state representation (e.g., ProPara). We thus frame the semantic comprehension of procedural text such as recipes, as fairly generic NLP subtasks, covering (i) entity recognition (ingredients, tools and actions), (ii) relation extraction (what ingredients and tools are involved in the actions), and (iii) zero anaphora resolution (link actions to implicit arguments, e.g., results from previous recipe steps). Further, our Recipe Instruction Semantic Corpus (RISeC) dataset includes textual descriptions for the zero anaphora, to facilitate language generation thereof. Besides the dataset itself, we contribute a pipeline neural architecture that addresses entity and relation extraction as well as identification of zero anaphora. These basic building blocks can facilitate more advanced downstream applications (e.g., question answering, conversational agents).

1 Introduction

Recently, several efforts have aimed at understanding recipe instructions (see Section 2). We consider such recipes as prototypical for procedural texts, for which processing is complex due to the need to (i) understand the ordering of steps (not unlike, e.g., event ordering in news), (ii) solve frequent ellipsis (i.e., zero anaphora) and coreference resolution, and (iii) track the state changes they involve (e.g., ingredients processed/combined to new entities). Especially the latter distinguishes procedural text processing from more traditional information extraction (e.g., from news).

Most existing works on recipes focus on recognizing pre-defined predicates, typically in the form of a limited set of instruction types (e.g., to convert the recipe to robot instructions) with predefined argument slots to fill. Further, they often rely on an available starting list of ingredients (which may not be available in other procedural text). Hence, current approaches towards recipe understanding make assumptions that are rather domain specific. In contrast, we aim for a more basic and generic structured representation of the procedural text, limiting domain-specific knowledge and building on more general semantic concepts. In particular, we build on semantic concepts as defined in PropBank (Kingsbury and Palmer, 2002), which are not domain-specific.

Note that our proposed form of structured representations not necessarily allows directly solving informational queries that require explicit reasoning and/or state tracking (e.g., "Where are the tomatoes after step 5?"). We however pose that properly detecting the various entities (e.g., ingredients and their derivations) and the actions that are executed on them (as described by verbs), with the appropriate coreference and zero anaphora resolution, would enable constructing a graph that facilitates such tracking. Thus, while our proposed representation based on the idea of joint entity and relation extraction (Bekoulis et al., 2018), provides useful input for it, such explicit state tracking and representation (e.g., as in ProPara, Dalvi et al., 2018) is left out of scope here.

In summary, this paper reports on our work-in-progress and makes two main contributions. First, we present our newly annotated Recipe Instruction Semantic Corpus (RISeC) dataset (Section 3), following the frame-semantic representation of PropBank (Kingsbury and Palmer, 2002). Since

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 821–826
December 4 - 7, 2020.

PropBank is domain-agnostic, the approach should be largely generalizable[1] to other procedural text settings. Second, we introduce a baseline framework (Section 4) to solve (i) entity recognition (ingredients, tools and actions), (ii) relation extraction (ingredients and tools linked to the actions), (iii) zero anaphora identification. Experimental evaluation thereof on RISeC is provided (Section 5).

2 Related work

From the perspective of structured representation, Tasse and Smith (2008) define the Minimal Instruction Language for the Kitchen (MILK), which is based on first-order logic to describe the evolution of ingredients throughout a recipe, and use it for annotation in the CURD dataset. Building on this effort, Jermsurawong and Habash (2015) extend CURD toward ingredient-instruction dependency tree parsing in SIMMR: they present an ingredient-instruction dependency tree representation of the recipe, but do not model instruction semantics. This contrasts with Maeta et al. (2015), who propose a pipeline framework for information extraction on Japanese recipes from the the recipe flow graph (r-FG) dataset (Mori et al., 2014). Maeta et al. use word segmentation, named entity recognition and syntactic analysis to extract predicate-argument structures and build a recipe flow graph that is conceptually similar to a SIMMR tree. Their work is conceptually closest to ours, in that they propose a chain of NLP subtasks (but we do not need word boundary identification in our English corpus). Yet, we build on a more elaborate and generic semantic relation scheme, PropBank (Kingsbury and Palmer, 2002). Further, methodologically we adopt neural network models as opposed to their logistic regression for NER and a maximum spanning tree (MST) parser for the relations (i.e., graph arcs). Tracking state changes is another key to understanding recipe language. Bosselut et al. (2018) predict the dynamics of action and entity attributes in recipes by employing a recurrent memory network. Their work includes sentence generation, but does not address the zero anaphora problem (see further) directly.

Besides recipes, other works focus on different procedural tasks. The ProPara[2] project aims at comprehending scientific processes and tracking the status of entities in them: Dalvi et al. (2018) focus on tracking entity locations (as well as their creation/destruction) using a specific matrix state representation (with a row per step, a column per entity). The proposed models however do not incorporate entity recognition and are specifically filling the chosen state representation. In our work, we rather stick to a more "basic" understanding, which is broader in scope than location tracking. In terms of datasets beyond the recipe domain, the work of Mysore et al. (2019) is noteworthy: it focuses on material synthesis and annotates domain-specific entities (materials, operations, conditions, etc.) and relations. The latter in our case are rather domain-agnostic (using PropBank).

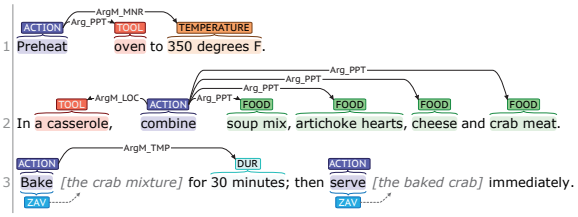

Figure 1: An annotated recipe. The fragments between brackets are manually added anaphora descriptions.

3 The RISeC Dataset

The following paragraphs describe our dataset and the annotations underlying the presented extraction task[3].

3.1 Dataset Collection

Recipes in our RISeC dataset are those from the SIMMR dataset.[4] Unlike SIMMR, we only use the instruction text of each recipe, and rather detect ingredients (as well as derived entities) from the text itself. We annotate the dataset using BRAT (Stenetorp et al., 2012), which eventually creates a directed acyclic graph where (i) vertices are *entities* (text spans) such as ingredients, tools, actions, intermediate products, and (ii) edges denote *relations* between entity spans. An example of our annotation is given in Fig. 1. Three expert annotators are involved in this task, who were are in close communication during the entire annotation process to maximize annotation consistency.

[1] While some of our entity types are specific to the cooking domain (e.g., "food", "temperature"), the relations that link action verbs to them are not (cf. PropBank).

[2] http://data.allenai.org/propara

[3] The annotated data is available for research at https://github.com/YiweiJiang2015/RISeC

[4] https://camel.abudhabi.nyu.edu/simmr/

3.2 Annotation Structure

Entity Types

`Action`: Most verbs, their present/past participles and verb phrases fall in this category. In addition to the Action label, specific verbs also carry a Zero Anaphora Verb (ZAV) label (see further).

`Food`: Ingredients, spices (salt, sugar, etc.), intermediate products (e.g., "the meat mixture"). If a sequence of ingredients is involved in an action, we label each of them individually, as in Fig. 1.

`Tool`: Appliances (e.g., oven), recipients (e.g., bowl), utensils (e.g., fork) used to perform an action involved in the cooking process.

`Duration`: Time interval for which an action lasts (e.g., '20 minutes', 'half an hour').

`Temperature`: E.g., "400 degrees F".

`Other`: This label is used for entities that cannot be attributed to any entity label above.

Further, we also annotate subclauses that provide information on certain actions as "entities". Thus, we abuse entity labeling to indicate them and thus limit their annotation to shallow parsing:

`Condition_Clause`: Sub-clauses led by conjunctions like "until", "till", "when", "before", usually expressing timing.

`Purpose_Clause`: Infinitives and sub-clauses led by for example "so that", "to make sure that".

Relation Types

Following the methodology of PropBank, we define a set of relations for the semantic roles in recipe instructions. These relations have the verb as origin and link an action to its arguments (`Arg_*`) or modifiers (`ArgM_*`). For details on their meanings, see PropBank's annotation guidelines (Babko-Malaya, 2005). However, to make the annotating schema self-consistent and adaptive to the cooking domain, we create (or extend) verb frames that are not (yet) included by PropBank. E.g., for the verb phrase "beat in", we borrow the argument structure from its main verb, i.e., "beat".

`Arg_PPT`: Participant, used for the argument which undergoes a change of state or is being affected by an action.

`Arg_GOL`: Goal, destination where an action ends.

`Arg_DIR`: Direction, the source where an action starts from. E.g., "Remove the pan from oven to a rack" where "oven" is `Arg_DIR` of the action "remove".

`Arg_PRD`: Predicate, used for the end product of an action. E.g., "Roll the cool dough into 3-inch ball" where the dough is transformed into "3-inch balls", `Arg_PRD` of the action "roll".

`Arg_PAG`: Agent, the subject that performs an action.

`ArgM_MNR`: Manner, describing how or in what condition we execute an action. E.g., in"Preheat the oven at 340 degrees", the relation `ArgM_MNR` links `Action` "preheat" to `Temperature` "340 degrees F".

`ArgM_LOC`: Location where an action takes place. This notion is not restricted to physical locations, but abstract locations are being marked as `ArgM_LOC` as well. E.g., in "Beat 2 eggs in the flour", `ArgM_LOC` links `Action` "beat" to `Food` "the flour"

`ArgM_TMP`: Temporal relation between action and timing nodes (Duration, Condition_clause).

`ArgM_PRP`: Purpose relation between action and purpose clause nodes.

`ArgM_INT`: Instrument, e.g., the utensil to accomplish the action.

`ArgM_SIM`: Simultaneous, linking two actions performed at the same time. E.g., in "Broil the lamb, moving pan so entire surface browns evenly", `ArgM_SIM` links "broil" to "moving".

Zero Anaphora Rephrasing

Zero anaphora is the phenomenon of implicit, unmentioned references to earlier concepts. Figure 1 gives two examples where explicit anaphors are manually added inside the brackets. The last sentence in Fig. 1 would be ungrammatical without the unmentioned "the crab mixture" and "the baked crab". In our annotations, we annotated 1,526 Zero Anaphora Verbs with candidate expressions for the zero anaphora, providing at least two alternatives: a succinct noun, as well as a more detailed noun phrase.

4 Model

We focus on two tasks: (1) joint entity recognition, relation extraction and zero anaphora identification, and (2) zero anaphora description generation. Next we present our models for each.

4.1 Entity recognition, relation extraction & zero anaphora identification

We use a span-based model, taking the input sequence of words as input, and passing it through 4 components: (i) word representation, (ii) span representation, (iii) entity recognition, and (iv) relation identification.

Word Representation: We use a BiLSTM as the

base encoder. The inputs are vector representations of the sentence tokens obtained by concatenating pre-trained GLoVe embeddings (Pennington et al., 2014) and character representations (using a CNN, ReLU and max pooling, as proposed by dos Santos and Guimarães, 2015). Further, we also experimented with pre-trained BERT models (Devlin et al., 2019) instead of Glove embeddings.

Span Representation: We enumerate all possible word spans from the input sentence and concatenate the aforementioned BiLSTM (h_{left}, h_{right}) encoder outputs at first (f) and last (l) end-point tokens of each span, together with its length (e_{len}) to obtain a span representation $(s_i = (h_{left,f}, h_{right,f}, h_{left,l}, h_{right,l}, e_{len}))$.

Entity Recognition & Zero Anaphora Verb Identification: We pass the selected span representations s_i through a feed-forward neural network (FFNN) yielding per-class scores for predicting entity types as well as binary Zero Anaphora Verb labels (with k entity classes, the FFNN thus has $k + 1$ outputs).

Relation Identification: The concatenation of two span representations (s_i, s_j) is passed through another FFNN to derive per-class relation scores. Since this is quadratic, we only pass the top 20% highest scored spans to the Relation FFNN: every span pair (s_i, s_j) is first passed through a pruning FFNN, and only its top-scored pairs are pushed through the Relation FFNN.

Training: For each recipe instance, the objective is to optimize the weighted sum of the negative log likelihood of span representation, entity classification and relation identification. We use Adam to optimize the model with learning rate 0.001.

4.2 Zero anaphora description generation

For the generation task, we build a baseline model corresponding to the sequence-to-sequence architecture used in Bahdanau et al. (2015). The input is the entire recipe, which we pass to an LSTM encoder taking the pre-trained GloVE embedding, concatenated with a binary label indicating whether it is a zero anaphora verb (ZAV), and (optionally) an entity type embedding if the token is of a given type. Since usually the target description that the decoder needs to generate is much shorter than the full recipe, we adopt bi-linear attention (Luong et al., 2015). The model is trained to minimize the negative log likelihood of an emitted token given the full input and predicted tokens.

	Glove	$\text{Bert}_{\text{base}}$	$\text{Bert}_{\text{large}}$
Entity	89.8	91.7	92.6
Zero Anaphora Verb	89.1	89.0	89.8
Relation	65.5	67.1	67.5

Table 1: Micro-F1 scores of models with Glove, $\text{Bert}_{\text{base}}$ and $\text{Bert}_{\text{large}}$ on the test set.

	Full	Test set		
	Count	Prec.	Recall	F1
Food	3,232	92.5	95.9	94.2
Action	3,061	96.6	97.4	97.0
Tool	1,138	92.9	86.8	89.8
Condition clause	487	93.0	71.1	80.5
Duration	411	85.7	87.4	86.5
Temperature	381	87.4	89.3	88.4
Other	270	54.2	34.7	41.9
Purpose clause	147	78.0	59.2	67.2

Table 2: Entity counts in full dataset and extraction results with $\text{Bert}_{\text{large}}$ on test set.

5 Experiments and results

We split our RISeC dataset into 50% training, 20% development and 30% test sets, using the same splits as SIMMR (Jermsurawong and Habash, 2015). We tune hyperparameters on the development set. Reported performance metrics are obtained on the test set.

In general, our span-based model shows good performance in the extraction task, as shown in Table 1. We obtain micro-F1 scores for the joint entity, zero anaphora verbs and relation identification tasks of respectively 89.8, 89.1 and 65.5 when using Glove word embeddings. With $\text{Bert}_{\text{large}}$ word encodings, performance consistently improves by 2.8, 0.7 and 2.0 percentage points respectively, indicating the applicability of the general linguistic knowledge from Bert on a cooking-domain task.

Individual entity and relation type performance is reported in Tables 2–3. As expected, Table 2 shows that entity F1 scores are positively correlated with the occurrence frequency, except for `Duration` and `Temperature`, of which the fixed pattern is easy to learn. The high precision and recall of important entities like `Food` and `Action` shows promising potential of our model for downstream applications like a question answering system in smart kitchen settings. The F1

		Full	Test set		
		Count	Prec.	Recall	F1
Argument Relations	Arg_PPT	3,196	94.1	69.3	79.8
	Arg_GOL	557	79.6	35.8	49.1
	Arg_DIR	91	93.9	34.5	50.4
	Arg_PRD	74	77.8	27.4	40.0
	Arg_PAG	25	0.0	0.0	0.0
Modifier Relations	ArgM_TMP	884	91.7	33.2	48.7
	ArgM_LOC	515	87.8	49.7	63.3
	ArgM_MNR	432	86.7	35.6	50.1
	ArgM_PRP	137	85.2	9.1	15.8
	ArgM_SIM	92	66.7	11.1	18.6
	ArgM_INT	73	77.4	20.3	31.8

Table 3: Relation counts in full dataset and extraction results with $\text{Bert}_{\text{large}}$ on test set.

scores of relation predictions in Table 3 show that the imbalanced distribution of relation types causes detection of several relations to be difficult, e.g., the low recall rates for `Arg_PAG` and `ArgM_PRP`. Future work should address this, e.g., using a larger dataset (or pretraining on non-recipe corpora).

While the detection of zero anaphora verbs (ZAV) performs well, our Seq2seq based description generation largely failed, with very low performance and oftentimes outputting the same descriptions (e.g., "mixture" or "chicken"). In hindsight, given the limited dataset size (order of 1.5k ZAV occurences in the full dataset) and the typically large training dataset needed for seq2seq models, this is not entirely unexpected. Further work on this task is clearly required.

6 Conclusion and Future Work

This paper introduced RISeC, a dataset for extracting structural information and resolving zero anaphora from unstructured recipes. The corpus consists of 260 recipes from SIMMR and provides semantic graph annotations of (i) recipe-related entities, (ii) generic verb relations (from PropBank) connecting these entities, (iii) zero anaphora verbs having implicit arguments, and (iv) textual descriptions of those implicit arguments. We reported on our work-in-progress with two baseline models using our corpus: (i) a neural span-based model extracting entities, zero anaphora verbs and relations, and (ii) a sequence-to-sequence attention model generating noun phrases for zero anaphora verbs.

We plan to continue working in this direction, making the dataset larger and more fine-grained, and especially, to investigate how it can be leveraged for human-machine interaction experiments.

Acknowledgments

The first author was supported by *China Scholarship Council* (201806020194). This research received funding from the Flemish Government under the *"Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen"* programme. We would like to thank anonymous reviewers who helped to improve the draft.

References

Olga Babko-Malaya. 2005. Propbank annotation guidelines.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA.

Ioannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. *ArXiv preprint arXiv:1711.05313*.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2018)*, pages 1595–1604, New Orleans, Louisiana.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jermsak Jermsurawong and Nizar Habash. 2015. Predicting the structure of cooking recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 781–786, Lisbon, Portugal. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1989–1993, Las Palmas, Canary Islands, Spain.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1412–1421, Lisbon, Portugal.

Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. 2015. A framework for procedural text understanding. In *Proceedings of the 14th International Conference on Parsing Technologies (IWPT 2015)*, pages 50–60, Bilbao, Spain.

Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 2370–2377, Reykjavik, Iceland.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Cícero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 102–107, Avignon, France.

Dan Tasse and Noah A Smith. 2008. SOUR CREAM: Toward semantic processing of recipes. *Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-LTI-08-005*.