Quantifying the contribution of different knowledge sources in narrative-based text understanding

Luc Steels[1] [0000−0003−0068−6551] and Lara Verheyen[2] [0000−0001−5191−3174]

[1] Barcelona Supercomputing Center, Barcelona, Spain
steels@arti.vub.ac.be
[2] VUB AI Lab, Brussels, Belgium
lara.verheyen@ai.vub.ac.be

Abstract. Narrative-based text understanding is the process of making a rich model of the situations being evoked by a text. This process goes significantly beyond coarse-grained natural language processing and even beyond the fine-grained linguistic analysis of a text based on lexicons and grammars. It requires integrating information gleaned from ontologies and common sense knowledge bases, logical inference, qualitative mental simulation, image interpretation (if the text is accompanied with images), discourse modeling, quantitative physical simulation, sensori-motor action and inclusion of extra-textual context. This paper introduces a novel way to integrate and measure the contributions from different knowledge sources in narrative-based understanding based on framing understanding as a process of raising questions, finding answers to questions, and interlinking questions. The proposed measures can be used to evaluate narrative-based understanding, evaluate the quality of texts, and provide a feedback signal for machine learning to improve the efficiency and efficacy of the understanding process. We illustrate the paper with an operational system that goes some way to understanding cooking recipes.

Keywords: narrative-based understanding · measures of understanding · cooking robots · human-centric AI · close reading

1 Introduction

There is a well-established distinction in literature studies between close and distant reading [8]. Distant reading refers to the use of methods from information retrieval, data mining and statistics to find information or regularities in texts without actually being concerned with meaning. Distributional semantics, word embeddings or counting token frequencies are all examples of tools enabling distant reading.

Close reading views a text as a narrative. Narratives explicate the events, agents and their roles, the temporal, spatial and causal relations, the intentions and motivations of actions, as well as the perspectives and values implied by the actions and the situations described in the text [3]. Close reading is about re-constructing the narrative underlying a text. This not only involves identifying the facts (the so called *fabula* or world model) but also how the facts were organised and framed to fit with a specific viewpoint (the so called *plot*) and how both the fabula and plot have been communicated via narrative texts, images or other types of semiotic objects (known as *narrations*) [1]. The construction or interpretation of narrations is in itself highly complicated, requiring the construction or reconstruction of linguistic models at several layers (lexical, morphological, syntactic, semantic, pragmatic).

Close reading is the first step in understanding. The second step, which happens concurrently with close reading, is to ground the narrative in the real world as experienced through a sensori-motor embodiment (for example if the text describes real world actions or gives instructions for actions in the world) and in past experiences stored in personal dynamic episodic and semantic memory [12]. When this grounding is achieved we speak about **narrative-based understanding** (Figure 1).

For example, when cooking a dish from a recipe, the cook has to construct a model which identifies the ingredients and the food manipulations in sufficient detail to effectively cook the recipe. During the cooking process the cook has to ground this narrative in the here-and-now of her kitchen. Ideally, a model of a recipe should also support the search for variations, for example to find alternatives for missing ingredients, or for alternative methods if the cooking process does not quite go the way it is described in the recipe [2]. As another example, consider the close reading of a book about the French revolution. The text stimulates the reader to build models of the key actors, their intentions and motivations, the salient events, the temporal and causal relations between these events and the social and governmental changes they cause. The reader has to integrate these facts and viewpoints with her own personal dynamic that contains narrative structures based on earlier readings or conversations and only if that successfully happens we can speak about understanding [13].

It is well known that close reading has to go significantly beyond the fine-grained processing of the text itself (which is already difficult to do because we need broad coverage precision grammars). Close reading requires many additional sources of knowledge to bear on the reconstruction of a narrative, simply because language is ambiguous and vague and gives only incomplete and fragmented information about the world and how it is viewed by the narrator. These additional sources include:

- Information about the **context**, for example, the kitchen with its various cabinets containing utensils and food ingredients, a refrigerator, etc.
- **Ontologies**, that come into two different types: **Conceptual ontologies** that define categories with their components and relations in terms of frames. For example, a frame for a cooking pot will specify what information should be kept about the pot, such as what its size and contents are or whether the pot should be covered with a lid or not. **Distributional ontologies** that provide embeddings derived from the statistical processing of co-occurrences in texts and can be used for approximative matching.
- **Common sense and domain knowledge** contains large numbers of axioms that can be used to infer additional facts not explicitly mentioned in the text.
- **Discourse models** specify what is in the focus of attention and which questions are important and unresolved. For example, when the list of ingredients contains ‘three tomatoes’, these tomatoes come into the focus of attention. Hence, when the recipe later mentions ’the tomatoes’, this description most likely refers back to these three tomatoes - and not to other tomatoes that are still in the pantry.

- **Sensori-motor grounding** is about the relation between categories and sensory inputs or motor actions. This is relevant when the text contains images or is describing actions that are being performed.
- **Qualitative or quantitative mental simulation** predicts either in a qualitative or quantitative way future or past world states. For example in a recipe the reader has to imagine the needs and effects of cooking actions because later steps will refer back to their outcome.

The grounding in the real world and in personal dynamic memory may also provide further feedback to constrain the possible interpretations coming out of close reading.

Just like humans, narrative-based understanding systems need to use every possible bit of information and every possible knowledge source as quickly as possible in order to arrive at the most coherent model that integrates all data and constraints. This process cannot be implemented as a linear pipeline where one algorithm feeds into another, as is common in data-driven AI, because of a paradox known as the *hermeneutic circle*: To understand the whole we need to understand the parts but to understand the parts we need to understand the whole [6].

Because of this paradox, narrative-based understanding has to be conceived of as a spiraling process. Starting from an initial examination of some input elements, for example a sentence with a lot of ambiguity, uncertainty and indeterminacy, hypotheses of the whole are constructed, which then provide top-down expectations to be tested by a more detailed examination of the same or additional sentences, leading to a clearer view of the whole, which then leads back to the examination of additional sentences, etc., until the narrative can be grounded in reality or episodic memory, reaching a state known as **narrative closure** [4].

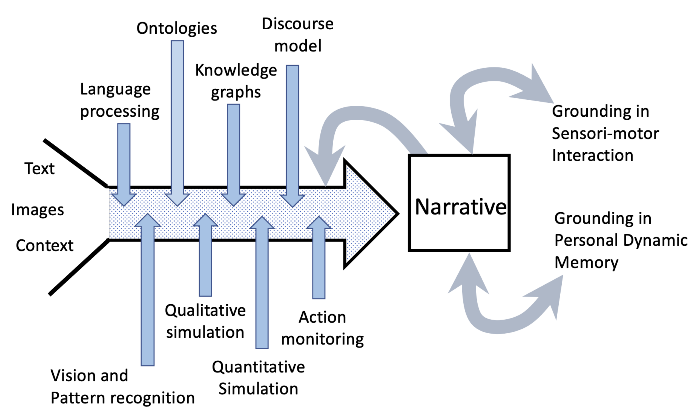

Fig. 1. Narrative-based understanding is the process of reconstructing the rich model underlying a text, using a variety of knowledge sources and strategies for grounding the narrative in sensori-motor interaction and personal dynamic memory.

This paper addresses the challenge of integrating many different knowledge sources in the service of understanding and quantifying progress towards narrative closure. We also want to measure how and when each knowledge source contributes to the understanding process.

The paper is illustrated with a concrete operational example from work on understanding recipes being developed by a team at the VUB AI Lab with Katrien Beuls and Paul Van Eecke. We focus here on the integration of context, language (lexicon and grammar), a discourse model, qualitative simulation and ontologies.

As illustration of this paper, we use one of the recipes that have been studied, namely a recipe for preparing almond cookies. It goes as follows:

> *Recipe for almond cookies:*
> Ingredients: 226 grams butter, room temperature. 116 grams sugar. 4 grams vanilla extract, 4 grams almond extract, 340 grams flour, 112 grams almond flour, and 29 grams powdered sugar
> Instructions:
> 1. Beat the butter and the sugar together until light and fluffy.
> 2. Add the vanilla and almond extracts and mix.
> 3. Add the flour and the almond flour.
> 4. Mix thoroughly.
> 5. Take generous tablespoons of the dough and roll it into a small ball, about an inch in diameter, and then shape it into a crescent shape.
> 6. Place onto a parchment paper lined baking sheet.
> 7. Bake at 175 degrees Celsius for 15 - 20 minutes.
> 8. Dust with powdered sugar.

The key novel idea proposed in this paper is to **view the understanding process in terms of questions and answers.** Each knowledge source can: (i) produce new questions, (ii) introduce answers to questions, (iii) constrain the answers to questions, or (iv) shrink the set of questions by realizing that the answers to two different questions are in fact the same. Narrative closure is reached when all critical questions have been answered, in particular the main question raised by the text.

The main question posed and answered by the Almond Cookies recipe is how to prepare almond cookies. This raises a host of other questions: What utensils are needed (a baking tray, a bowl, ...), where can things be found or put in the kitchen (freezer, pantry, ...), what ingredients are necessary (116 grams of sugar, 4 grams almond extract, ...), which objects need to be prepared (a mix of flour and almond flour, a small ball of dough, ...), which actions need to be performed (add flour, bake, ...), and properties of all these entities and actions.

We do not discuss in this paper how each knowledge source is implemented. Full detail is provided in a webdemo available here: https://ehai.ai.vub.ac.be/demos/recipe-understanding/) and in other papers referenced in the text.

2 Questions and answers

The question-driven understanding framework is operationalized as follows:

Questions are operationalized as variables. Following AI tradition, the name of a variable is written as ?*variable-name* where the variable-name is a symbol that is chosen on the basis of the type of object that is an answer to the question. This is meant to be meaningful for us. Variable names are formal symbols that obtain their function only from their connections to other symbols not from the name they happen to have. Variable names typically have subscripts, as in ?bowl-1, ?bowl-2, ... , which refer to different bowls in the kitchen.

Answers are operationalized as bindings between variables and identifiers of entities. A binding has a score reflecting the certainty with which the entity is estimated to be the right answer. It also has a moment on a timeline from when the entity is available for further actions. Some actions discard entities (like bowls) after usage or the entities disappear because they are now indivible part of other entities. On the other hand particular actions, such as boiling water, may take a certain amount of time and hence the pot with boiled water is only in existence from a future moment in time. The handling of time is a complex extensively researched subject in qualitative mental simulation [5] and its discussion is beyond the scope of this paper.

Entities are objects, events or (reified) concepts. They are designated with a symbol, called the entity-id, but now without a question mark and with angular brackets. The name is again indicative of the type of entity being named and we use subscripts to distinguish between different entities of the same type, as in <butter-331> or <bowl-710>. Entities either refer to real world observational data, for example a region in an image, to virtual entities that may or may not exist in reality, or to entities in a knowledge graph in which case we use the URI (Universal Resource Identifier) as unique identifier.

Entities may have different states, for example butter can be solid or become fluid when melted. To represent this, an entity has not only a persistent id but also different temporal existences, each marked with its own identifier by the use of additional subscripts. For example, <butter-331-1> with the persistent id <butter-331> might change after heating into <butter-331-2> with the same persistent-id but different properties.

Descriptions of entities are stored in the world model in the form of frame instances. The possible frames and their slots are defined as part of the conceptual ontology. The ontology typically also defines default values for these slots. When a frame is used to describe a particular entity or set of entities, it is first instantiated.

Frames and instances of frames are designated by symbols with square brackets. Names of instances have indices. In the recipe example, there is for example a frame for [bowl] with slots for the bowl itself, the contents, the size, the cover, whether the bowl has been used, etc. A specific bowl entity, e.g. <bowl-75>, is described by a frame instance, e.g. [bowl-75] and the name of the entity is stored in its persistent-id slot.[3]

In the current implementation, the world model is implemented in the style of object-oriented and frame-based knowledge representation systems and uses CLOS (the Common LISP Object System) [7] as implementation medium.

Constraint Networks. The meaning expressed by a text fragment (for example a sentence) translates into a constraint network that links questions (i.e. variables) to constraints for finding answers. It is implemented using the IRL constraint system developed for grounding language in embodied robotic interactions[9]. IRL is itself work-

[3] All these indices are of course automatically constructed by the understanding system itself.

ing on top of CLOS. A constraint performs computations to find answers to questions knowing the answers to others. Solving a constraint could involve creating new frame instances for additional entities, creating a new world state, copying an existing state with temporal changes, calling up other knowledge sources such as a quantitative simulation or consultating semantic web resources, etc. The constraint solver iterates through the constraints progressively filling in as much information as possible. Constraint solving is a search process because the same question may potentially have multiple answers until this is resolved by the application of more constraints. Moreover some questions may remain unanswered at the end of considering a constraint network, until more information arrives later.

Discourse Model. The case study contains a simplified discourse model which contains a set of accessible entities and their frames, together with the questions they were an answer to (called the binding-variable). An entity becomes one of the accessible entities if it has been mentioned in the text or is the result of some action. It remains in the focus of attention until it has been referred to later or used in another action.

3 A First Example

To illustrate understanding as a process of questions and answers we now look at some very small fragments of the interpretation process for the Almond Cookies recipe. The full interpretation process for (an earlier version of) this example is shown in the web demonstration accessible through this link: https://ehai.ai.vub.ac.be/demos/recipe-understanding/.

Contributions from context

The process starts with the instantiation of a particular kitchen model (called the kitchen-state) as the initial world model. This kitchen-state then becomes an accessible entity in the discourse model. The kitchen-state contains entities for an empty countertop, an empty oven, a freezer, a fridge with a bowl of butter, a pantry and a kitchen-cabinet. The pantry and cabinet contain various medium and large-sized bowls, whisks, a baking tray, and baking paper. For each of these entities frames are instantiated and added to the world model, based on the kitchen ontology. An example of the frame for the fridge instance [fridge-2-1] in the initial kitchen-state is shown in Figure 2, left. This step in processing generates the first question, namely the accessible entity `?kitchen-state-1`.

Contributions from language

Next the list of ingredients is parsed, generating a set of questions and initial answers. We look only at the first ingredient specification namely

“226 grams butter, room temperature."

The lexicon and grammar in this experiment uses a construction grammar, more specifically Fluid Construction Grammar (FCG) [10], [11], which maps syntactico-semantic patterns to hierarchical grammatical structures and meanings. In this case, the *Quantity-Unit-Ingredient-Cxn* matches with the linguistic units for the phrase “226 grams butter", where the quantity here is expressed with the numeral “226", the unit with the noun “grams", and the ingredient with the noun “butter". The linguistic structure produced after application of the construction is shown in Figure 2, right.

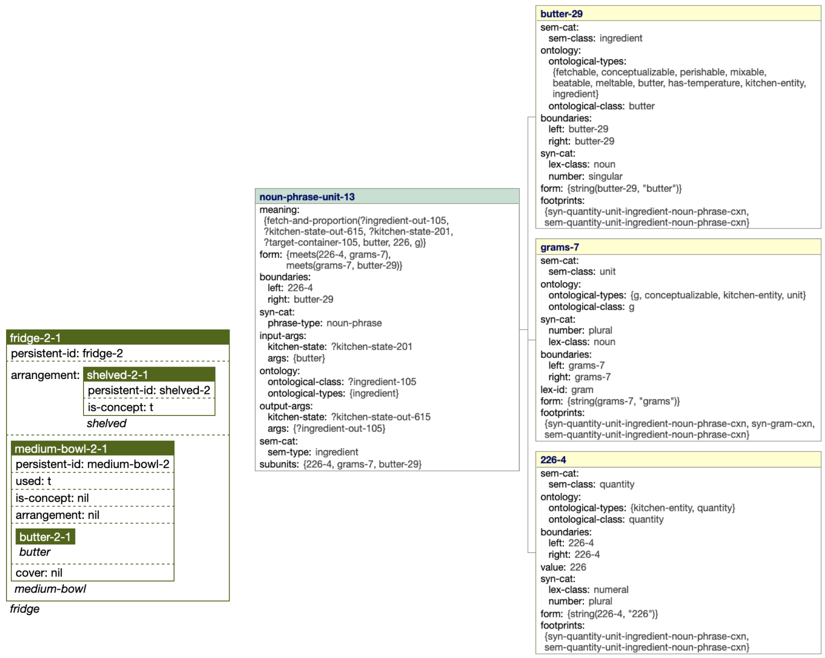

Fig. 2. *Left:* Description of fridge entity based on the initial context. There is a specific arrangement of the items in the fridge stored in the `arrangement` slot filled with the [shelved-2-1] frame instance. The contents contain the [medium-bowl-2-1] frame instance which contains itself a [butter-2-1] frame instance. *Right:* Syntactic and semantic structure for '226 grams butter'. The images shows linguistic units like noun-phrase-unit-13 or butter-29 together with syntactic and semantic properties. In FCG all these properties, such as forms, word order, grammatical categories, ontological types of referents, etc., are all described explicitly in terms of feature structures. The footprints feature in a linguistic unit indicate which constructions have contributed to the creation and completion of a linguistic unit.

The construction adds the following questions and constraints to the constraint network (stored in the meaning slots of the different units):

```
(bind quantity ?quantity-90 226)
(bind unit ?unit-91 grams)
(bind ingredient ?ingredient-92 butter)
(fetch-and-proportion ?ingredient-out-6 ?kitchen-state-out-35
   ?kitchen-state-3 ?target-container-6 ?quantity-90 ?unit-91
   ?ingredient-92)
```

The `bind` constraint has three arguments, a type (for example quantity), a question (for example `?quantity-90`), and a value or ontological class (for example 226). The first three `bind` constraints introduce not only three questions but also three answers (resp. for the quantity, the unit and the ingredient). These answers are coming straight from the sentence and count as contributions from language.

The last constraint, `fetch -and-proportion`, introduces four additional questions which can only be answered by applying the constraint (discussed later). So the total after parsing “226 grams butter" is 7 questions added and 3 answered. Parsing the remainder of the sentence “, room temperature" triggers the *Ingredient-Quantity-Unit-Cxn* which adds the following additional constraints to the constraint network:

```
(bind unit ?unit-100 degrees-celsius)
(bind quantity ?unit-quantity-99 18)
(bring-up-to-temperature ?ingredient-at-room-temperature-99,
   ?kitchen-state-690, ?kitchen-state-out-586, ?ingredient-in-99,
   ?quantity-99, ?unit-100)
```

We see six questions being added and two are already answered. But the grammar can do more because the *Ingredient-Quantity-Unit-Cxn* can link differerent questions as having the same referent, namely the ingredient described by “226 grams butter" (`?ingredient-out-98`) is the same as the ingredient which is input for the bring-up-to-temperature constraint (`?ingredient-in-99`) and the kitchen-state when starting to bring the temperature of this ingredient up to date (`?kitchen-state-in-522`) is the same as the kitchen state from fetch-and-proportion (`?kitchen-state-out-586`). So the number of questions gets reduced from 4 to 2 through the grammar. Figure 3 shows all the questions currently active as well as the answers.

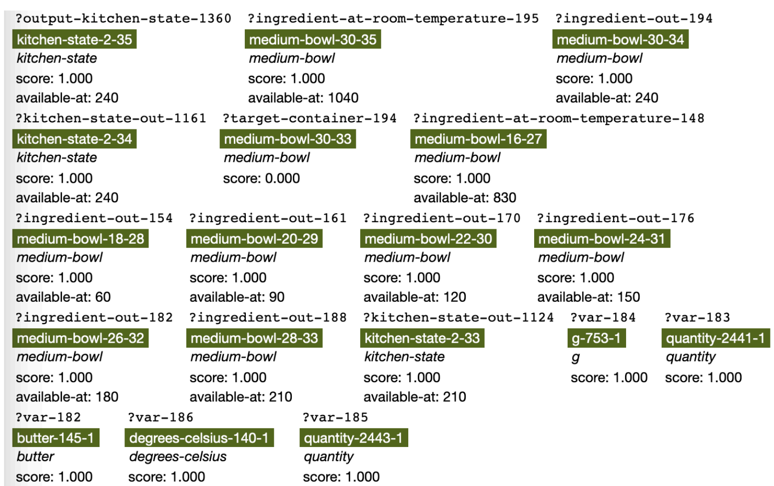

Fig. 3. Questions introduced after handling the first ingredient specification “226 grams butter, room temperature." and answered through language processing, the discourse model and qualitative mental simulation. Note that a binding has a score and also a time-moment when it becomes available. The latter is relevant in qualitative mental simulation.

Contributions from the Discourse Model

The discourse model is consulted while parsing the sentence and before the qualitative mental simulation. Recall that the initial kitchen-state (bound to `?kitchen-state-1`) was one of the accessible entities in the discourse model and thus a question that needs to be solved. The *Ingredient-Quantity-Unit-Cxn* requires a binding for a kitchen-state which it finds in the accessible entities. So this question from the discourse model is now answered.

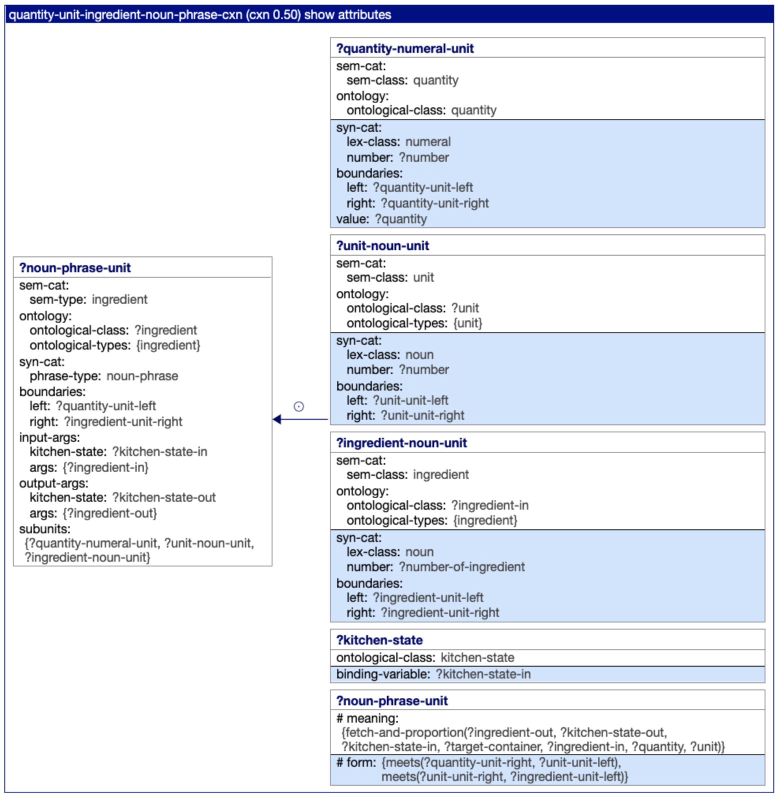

Fig. 4. The *Ingredient-Quantity-Unit-Cxn* looks during parsing for the kitchen-state, and a sequence of linguistic units (words in this case) introducing a quantity, a unit, and an ingredient respectively. It then builds a noun-phrase which introduces the `fetch-and-proportion` constraint.

Contributions from Qualitative Mental Simulation

Qualitative mental simulation (called qualitative physics in earlier AI work [5]) means to execute the effect of physical actions in a qualitative model of the world. For example, to simulate ‘fill the pot with water’ we go from a world state in which there is a pot which is empty to a state where it is filled with water. There are many things we do not know because they have not been specified, for example, how big the pot is, how long it took the fill it, whether the pot was filled to the top or not. Nevertheless we can continue to reason or understand subsequent sentences that involve the pot, like ‘boil the water’.

Qualitative mental simulation is done here through the IRL constraint system. A constraint in IRL has associated methods that specify the bindings of certain variables given other variables and the state of the objects involved after the action has been undertaken. For example, the `fetch-and-proportion` constraint has a method that finds in the current kitchen-state an ingredient (in this case the butter), finds an unused bowl in the cabinet, puts it on the kitchen's countertop, weighs and takes a slice of the butter, and puts it in the bowl. When this method is invoked, the necessary entities are created or their properties are changed and the relevant variables are bound to these entities. Other actions happen with the `bring-up-to-temperature` constraint. The net result of all questions and answers so far are shown in Figure 3. All questions have been answered.

Contributions from the Ontology

The conceptual ontology defines slots for the newly created entities based on the frames with which they are defined. For example, the butter is in a solid state but when it is brought up to temperature, a new butter entity is created (with the same ID as the previous one) and added to the current kitchen-state. The various slots and values for these slots are questions contributed by the ontology and answers supplied during mental simulation. They are not shown in Figure 3 for space limitations. Moreover when collecting statistics, we only count the slots that have unknown values and defaults that are overriden by other knowledge sources. In a similar way, all the other ingredients are fetched and then proportioned in a bowl on the countertop.

4 A Second Example

We now look at a second example, namely the first instruction of the recipe:

"Beat the butter and the sugar together until light and fluffy".

From a linguistic point of view we find here a *Resultative-Goal-Cxn* with a resultative-transitive clause ("beat the butter and the sugar together") and a goal-state ("until light and fluffy"). The transitive clause is an imperative and itself the product of a *Resultative-Cxn* with the pattern [action object *and* object result]. The action here is introduced by the imperative verb "beat", the first object is given by the noun-phrase "the butter", the second object by the noun-phrase "the sugar", and the result expressed by the adverb "together".

The various constructions build a constraint network with three constraints: two that transfer the contents of the bowl containing the butter `<medium-bowl-16-3>` and the bowl that contains the sugar `<medium-bowl-18-4>` to a new bowl `?input-container-96` and one for the beat-operation:

```
(transfer-contents ?output-container-?x-21 ?rest-x-42
  ?output-kitchen-state-x-21 ?kitchen-state-out-75
  ?empty-container-21 ?ingredient-at-room-temperature-7
  ?quantity-x-42 ?unit-x-42)
(transfer-contents ?input-container-96 ?rest-y-42
  ?input-kitchen-state-115 ?output-kitchen-state-x-21
  ?output-container-?x-21 ?ingredient-out-13
  ?quantity-y-42 ?unit-y-42)
```

```
(beat ?output-container-115 ?output-kitchen-state-134
  ?input-kitchen-state-115 ?input-container-96 ?tool-58)
```

Based on these constraints, the IRL constraint solver finds a binding for each of the variables, thereby providing an answer to the questions. The total set of questions and answers is shown in Figure 5.

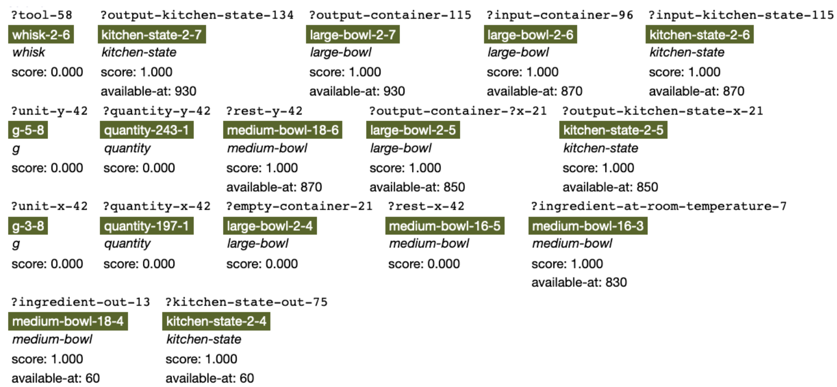

Fig. 5. The different questions and answers after using language, the discourse model, the ontology, and the qualitative mental simulation for the sentence “Beat the butter and the sugar together until light and fluffy".

Contributions from the Discourse Model

Although the other knowledge sources (in particular the qualitative mental simulation and the ontology) provide many of the answers, we examine in more detail only the utilisation of the discourse model. The sentence refers to “the butter" and "the sugar". Linguistically speaking these are both instances of the *Def-Determiner-Noun-Cxn* where “the" is a definite article and the Noun is here “butter" and “sugar" respectively. The meaning of a definite article is to look into the discourse model for an entity that is of the same ontological class as introduced by the noun. In this case such entitites can be found in the discourse model as acccessible entities introduced by earlier phrases and so this way answers for `?ingredient-out-13` and `?ingredient-at-room-temperature-7` can be found. As before the kitchen-state is found in the current discourse model.

5 Measures

Given the framework of questions and answers we can now define measures characterizing the understanding process. Specifically we can at each timepoint determine the number of questions introduced and answered and the knowledge sources responsible them. We can also determine the percentage each knowledge source has contributed so far. In what follows we show the output of these measures applied to the complete Almond Cookie recipe given earlier.

When close reading the Almond Cookies recipe, a total of 344 questions were asked (not counting the slots in the different frames that have default values, there are about 400 of them). All of these questions are answered at the end of the execution process, thereby reaching narrative closure. Language is responsible for introducing 159 questions and solving 47 questions. The discourse model introduces 31 questions and solves 62 questions. Only one question rises from mental simulation, while it solves 120 questions. The ontology introduces a total of 153 questions and provides 115 answers.

It is clear that the questions that are posed by a certain knowledge source are not necessarily solved by the same knowledge source. The questions that were raised by language (in total 159 questions) are partly solved by the language system itself (47 answers) and by the discourse model (31 answers) which is active during language processing. The remaining questions from language are solved by the mental simulation (81 answers). The discourse model raised 31 questions which are all solved by the discourse model itself. The ontology asks 153 questions which are partly solved by the ontology itself by the use of defaults (115 answers). The other 38 questions are solved by the mental simulation.

Figure 6 shows the contributions of the different knowledge sources to answering the questions during the execution of the recipe. The total number of questions that come up during the language processing and the mental simulation are shown. You can see that the number of questions increase during the recipe as well as the answers. At the end, the number of answers are equal to the number of questions implying that narrative closure has been reached.

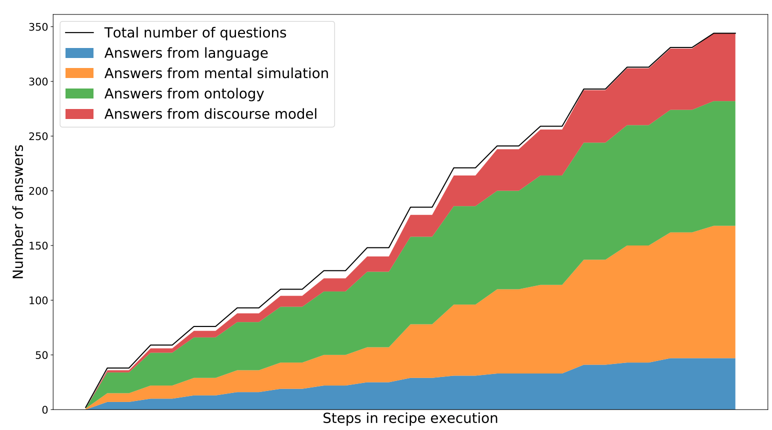

Fig. 6. Graph showing the contribution of different knowledge sources, the number of questions that arise and the number of available answers.

The next plot (Figure 7) shows the remaining questions after each step in the execution of the recipe. At the beginning of the recipe, there are no remaining questions. Then, the number of remaining questions increases during the ingredient instructions. After the ingredients have been processed the cooking instructions are handled and the number of remaining questions progressively decreases. At the end, no open questions remain.

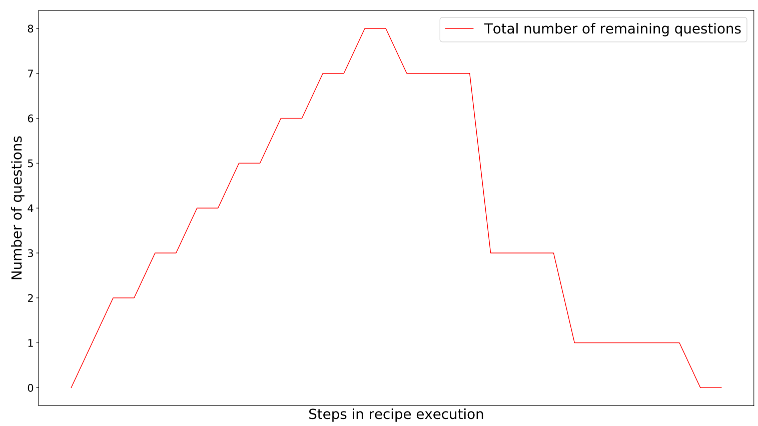

Fig. 7. Graph showing the evolution of the remaining questions during the execution of the recipe.

Figure 8 shows in more detail how many answers the different knowledge sources provide at two moments during the reading of the recipe. The first pie chart shows the contributions of the knowledge sources to the answers after processing the instructions related to the ingredients. The second is at the end of the recipe. After the ingredients list, there are 2.6% remaining questions. These questions are mostly from the discourse model and represent the ingredients that are fetched and that are waiting to be used later in the recipe. At the end of the recipe, the percentage of the remaining questions reaches 0%, meaning that all questions have been solved. The language also gradually becomes a smaller contributor to answering the questions, as mental simulation and the discourse model relatively become more important. Indeed, the mental simulation becomes an important provider for answers while instructions get more complex. In the beginning, the ontology is the largest contributor (46.6%). This is due to the defaults that are used in the beginning of the recipe. Later defaults are less used as mental simulation provides answers to the questions of the ontology.

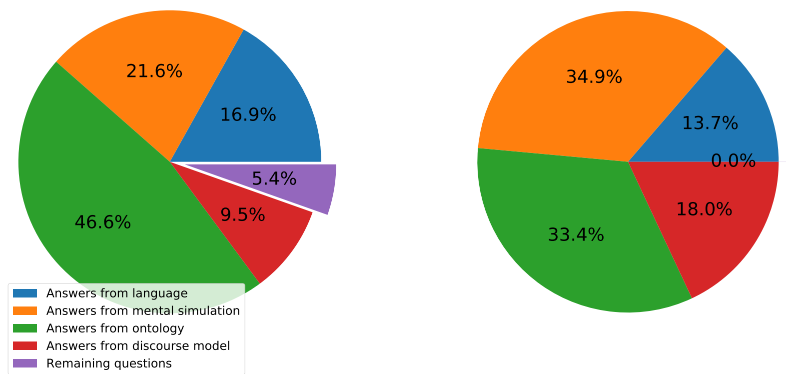

Fig. 8. Measurements of answered questions at two moments in the close reading of the recipe. The total amount of the pie chart is the total number of questions that were raised until that moment. The different sections denote the different knowledge sources. Over the course of the recipe, the number of remaining questions declines.

6 Conclusions

The main contribution of this paper is a system for integrating different knowledge sources in the service of building a narrative. We have been working on a concrete implemented system that performs this task using a framework of questions and answers. Different knowledge sources generate questions and the same or other knowledge sources come up with answers for these questions.

Given this framework we then explored ways to track and measure the addition, reduction or answering of questions by different knowledge sources. More concretely, we focused on the use of ontologies, language, discourse models and mental simulation.

The work presented here is just a first tiny step in building a quantitative infrastructure for tracking and evaluating understanding in AI systems. There are still several other quantitative measures that we have defined and will be discussed in forthcoming papers. One of them is connectedness. It measures how the different constraint networks introduced by each phrase in the recipe connects to other networks. If narrative closure occurs, all constraint networks become one large fully-connected network.

Having quantitative measures is useful to pin down precisely the contribution of a particular knowledge source or to provide feedback to the attention mechanism that guides what knowledge sources should preferentially be used or what areas of a narrative network should be the focus of attention. Quantitative measures can also play a role as feedback signal for improving the efficiency and efficacy of understanding.

In ongoing work we integrate additional knowledge sources (in particular quantitative mental simulation), scale up the knowledge sources used so far (for example scale up the grammar and the ontology), apply the framework to many additional recipes, and use other measures.

7 Acknowledgement

This paper was funded by the EU-Pathfinder Project MUHAI and the authors thank the host laboratories for this work: the Venice International University (LS) and the VUB AI lab (LV). We thank Paul van Eecke and Katrien Beuls from the VUB AI Laboratory for laying the basis for the almond cooking case study used here and Remi van Trijp for initial discussions on operationalizing the question-answer framework . The experiment is part of the EU Pathfinder project MUHAI on Meaning and Understanding in Human-Centric AI. Lara Verheyen is funded by the 'Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen' of the Flemish Government.

References

1. Bal, M., Boheemen, C.V.: Narratology: Introduction to the theory of narrative. University of Toronto Press, Toronto, Canada (1997)
2. Beetz, M., Jain, D., Mösenlechner, L., Tenorth, M., Kunze, L., Blodow, N., Pangercic, D.: Cognition-enabled autonomous robot control for the realization of home chore task intelligence. Proceedings of the IEEE **100**(8), 2454–2471 (2012)
3. Bruner, J.: The narrative construction of reality. Critical Inquiry **18(1)**, 1–21 (1991)
4. Carroll, N.: Narrative closure . Philosophical Studies. **135**, 1–15 (2007)
5. Forbus, K.: Qualitative Physics: Past, Present and Future., pp. 239–296. Exploring Artificial intelligence, Morgan Kaufman (1988)
6. Gadamer, H.G.: Hermeneutics and social science. Cultural Hermeneutics **2(4)**, 207–316 (1975)
7. Kiczales, G., des Rivieres, J., Bobrow, D.: The Art of the Metaobject Protocol. The MIT Presss, Cambridge Ma (1991)
8. Moretti, F.: Conjectures on world literature. New Left Review **1** (2000)
9. Spranger, M., Pauw, S., Loetzsch, M., Steels, L.: Open-ended Procedural Semantics., pp. 159–178. Language Grounding in Robots., Springer-Verlag, New York (2012)
10. Steels, L.: Computational Issues in Fluid Construction Grammar., Lecture Notes in Computer Science, vol. 7249. Springer Verlag, Berlin (2012)
11. Steels, L.: Basics of fluid construction grammar. Constructions and Frames **2**(9), 178–225 (2017)
12. Steels, L.: Conceptual foundations for human-centric ai. In: Chetouani, M., Dignum, V., Lukowicz, P., Sierra, C. (eds.) Advanced course on Human-Centered AI. ACAI 2021. vol. LNAI Tutorial Lecture Series. Springer Verlag (2022)
13. van Trijp, R., Blin, I.: Narratives in historical sciences. In: Steels, L. (ed.) Foundations for Incorporating Meaning and Understanding in Human-centric AI. MUHAI consortium (2022)