

# Semantic parsing with fuzzy meaning representations

**Pavlo Kapustin**

University of Bergen  
pavlo.kapustin@uib.no

**Michael Kapustin**

Moscow Institute of Physics and Technology  
michael.kapustin@gmail.com

## Abstract

We propose an approach and a software framework for semantic parsing of natural language sentences to discourse representation structures with use of fuzzy meaning representations such as fuzzy sets and compatibility intervals. We explain the motivation for using fuzzy meaning representations in semantic parsing and describe the design of the proposed approach and the software framework, discussing various examples. We argue that the use of fuzzy meaning representations have potential to improve understanding and reasoning capabilities of systems working with natural language.

## 1 Introduction

The meaning representation based on fuzzy sets was first proposed by Lotfi Zadeh (Zadeh, 1971; Zadeh, 1972). One of the very interesting properties of this representation is allowing to quantitatively describe relations between different concepts (e.g. “young”/“age”, “common”/“surprisingness”, “sel-dom”/“frequency”), as well as represent vagueness and imprecision that are so common to natural language. We recently proposed a related meaning representation, compatibility intervals, that, instead of using membership functions, describes similar relations using several intervals on a certain scale (Kapustin and Kapustin, 2019a).

Fuzzy meaning representations are relatively little known among linguists, and little used in natural language processing (Carvalho et al., 2012; Novák, 2017). We believe that this is well explained by the fact that these rich interpretations are generally not easy to learn from data, compared to the representations like word embeddings that can be derived from text. However, we believe that due to their expressiveness these representations have a huge potential when it comes to understanding and reasoning capabilities of the systems working with natural language. They allow to quantitatively express differences between similar language constructs (synonyms/antonyms, stronger/weaker constructs, wider/narrower constructs, etc.), making this knowledge available to the systems. In addition, compared to more opaque representations like word embeddings, they are also linguistically interpretable, potentially allowing to build the systems whose behaviour can be easier analyzed and explained.

We would like to contribute to the knowledge of using fuzzy meaning representations in natural language understanding, in particular, in semantic parsing. In this introductory paper we describe an approach and a software framework for semantic parsing of natural language sentences to discourse representation structures (DRS) with the use of fuzzy meaning representations.

First, we use these representations to enrich the resulting DRS with certain types of semantic information that can be helpful for downstream applications. In particular, both meaning similarity and vagueness can in many cases be directly represented in the DRS with the use of fuzzy meaning representations. For example, in the suggested approach, the constructs “this was not completely expected” and “this was fairly surprising” should lead to similar parses, expressing the meaning of the words “expected” and “surprising” in terms of the same underlying properties. Constructs “not completely” and “fairly” are modeled as modifiers acting on the words “expected” and “surprising” and transforming their

---

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

underlying representations (either the fuzzy sets or the compatibility intervals). This allows the systems to analyze the similarity of the constructs computationally.

Second, fuzzy meaning representations also play a key role during the semantic parsing, as they are used for the assessment of the understanding level and scoring of the different interpretations.

## 2 Related work

### 2.1 Fuzzy meaning representations

In his early works, Lotfi Zadeh suggests modeling meaning of certain types of adjectives (e.g. “small”, “medium”, “large”) as fuzzy sets, and some linguistic hedges (e.g. “very”, “slightly” — as operators, acting on these fuzzy sets (Zadeh, 1971; Zadeh, 1972). He introduces the concept of a linguistic variable (a variable whose values are words or expressions in a natural language) and suggests that values of membership function can be seen as degrees of compatibility between the value of the function argument and the construct the membership function is describing (Zadeh, 1975; Zadeh, 1978). Novák (2017) describes Fuzzy Natural Logic, a mathematical theory attempting to model the semantics of natural language, including Theory of Evaluative Linguistic Expressions (Novák, 2008), and further studies the concept of the linguistic variable (Novák, 2020).

Hersh and Caramazza (1976) introduce logical and linguistic interpretations of membership functions. We discuss these interpretations and other issues related to modeling natural language constructs with fuzzy sets (Kapustin and Kapustin, 2019b) and introduce another another fuzzy meaning representation, compatibility intervals (Kapustin and Kapustin, 2019a), also conducting a small-scale experiment that relates some language constructs to compatibility intervals (Kapustin and Kapustin, 2020).

There is some work aiming to make fuzzy sets easier to learn from data. Runkler (2016) describes an approach for generation of linguistically meaningful membership functions from word vectors. We believe that compatibility intervals (Kapustin and Kapustin, 2019a), a somewhat simpler representation than the fuzzy sets, may also be easier to learn from data.

Earlier we described a theoretical framework (Kapustin and Kapustin, 2015) for computational interpreting of natural language fragments, suggesting modeling meaning of words as operators. Some of the ideas of this framework are tested in a simplified setting in Kapustin (2015). Our approach and software framework also draw upon many of the ideas from that work.

### 2.2 Semantic parsing

There has been a growing body of research in semantic parsing with use of different meaning representations, including Minimal Recursion Semantics (Copestake et al., 2005), Discourse Representation Theory (Kamp and Reyle, 2013), Abstract Meaning Representation (Banarescu et al., 2013), Broad-Coverage Semantic Dependencies (Oepen et al., 2014; Oepen et al., 2015), Universal Conceptual Cognitive Annotation (Abend and Rappoport, 2013) and Universal Decompositional Semantics (White et al., 2016).

Abzianidze et al. (2019) describe a highly expressive scoped meaning representation based on Discourse Representation Theory (Kamp and Reyle, 2013) that combines logical (negation, quantification and modals), pragmatic (presuppositions) and lexical (word senses and thematic roles) aspects of semantics. The Parallel Meaning Bank (Abzianidze et al., 2017) uses this representation to annotate a corpus of translations including over 11 million words in four different languages. In the Parallel Meaning Bank concepts, states and events are annotated by the word senses from WordNet (Fellbaum, 1998), relations are modeled with thematic roles from VerbNet (Bonial et al., 2011) and optionally annotated with semantic roles from FrameNet (Baker et al., 1998). In addition to CCG categories (Steedman, 2000), the Parallel Meaning Bank annotates the words with semantic tags (Abzianidze and Bos, 2017). In our work, we loosely follow Abzianidze et al. (2019), extending it with fuzzy meaning representations.

### 3 Background

#### 3.1 Fuzzy meaning representations

Using fuzzy sets, one may model meaning of language constructs by relating them to a certain property that is described by these constructs (e.g. “age”, “time”, “frequency”, etc.). Then, the language construct is represented as a fuzzy set, and the membership function of this set models compatibility of the construct with different values that the property may take. For example, earlier we suggested how one could relate the meaning of depicted words to properties “surprisingness” and “frequency” by using one-dimensional *projections*, see figs. 1 and 2 (Kapustin and Kapustin, 2019b).

Note that on fig. 2  $\mu_{\text{seldom}}(0)$  and  $\mu_{\text{often}}(1)$  are significantly lower, than 1. This corresponds to the *linguistic interpretation* (Hersh and Caramazza, 1976; Kapustin and Kapustin, 2019b) that is related to scalar implicatures and models the fact that stronger words like “never” and “always” are more likely to be used near the ends of the scale, than words like “seldom” and “often”.

We recently proposed compatibility intervals, a meaning representation closely related to fuzzy sets (Kapustin and Kapustin, 2019a). Compatibility interval is an interval of property values on some scale that are compatible with a given language construct. Compatibility intervals consist of the main subinterval with high compatibility, and optional left (“increasing”) and right (“decreasing”) subintervals adjacent to the main subinterval. The following invariants are maintained: all the values in the main subinterval have equal (high) compatibility, and the closer the values are to the main subinterval the higher their compatibility is.

Consider an example of how one could define compatibility intervals for different age groups (which is, of course, entirely subjective). We use double hyphens between the start and the end of the main subinterval, and single hyphens between the start and the end of the left and the right subintervals).

```
child: [0 -- 15-20]
young1: [0 -- 30-50]
young2: [0-18 -- 30-50]
adult: [15-20 -- 100]
middle-aged: [40-45 -- 65-70]
old: [70-80 -- 100]
```

Note that construct *young<sub>1</sub>* corresponds to the *logical interpretation*, modeling the fact that newborns and infants are as young as one can be, while construct *young<sub>2</sub>* corresponds to the *linguistic interpretation*, modeling the fact that when someone refers to a person as “young”, we usually do not imagine a newborn or an infant.

Let’s consider how one could approximately translate figs. 1 and 2 to compatibility intervals:

```
expected: [0 -- 0.1-0.2]
common: [0-0.1 -- 0.2-0.3]
possible: [0-0.2 -- 0.5-0.6]
extraordinary: [0.6-0.9 -- 1]

seldom: [0-0.1 -- 0.2-0.3]
occasionally: [0-0.2 -- 0.4-0.8]
regularly: [0-0.4 -- 0.7-1]
often: [0.6-0.8 -- 0.9-1]
```

As long as both fuzzy sets and compatibility intervals model compatibility between a language construct and different values a property may take, and they both directly represent vagueness, we refer to both as fuzzy meaning representations and consider them somewhat interchangeable for the purposes of the paper. Although, there are some differences. While compatibility intervals is a simpler representation that may be easier to work with (Kapustin and Kapustin, 2019a) and may also be easier to learn from data, we consider fuzzy sets to be a more general, and in some sense, more powerful representation. For example, one cannot use compatibility intervals to model situations that use multi-dimensional membership functions (Kapustin and Kapustin, 2019b).

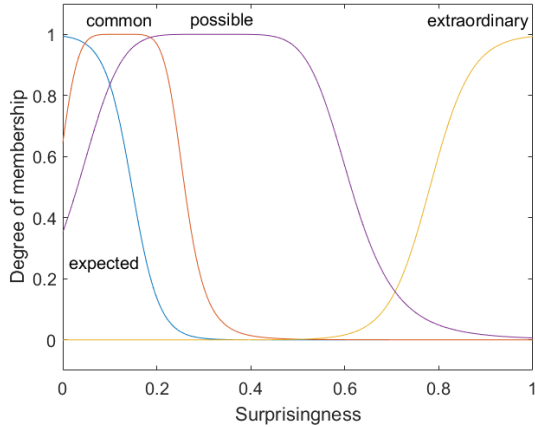


Figure 1: Depicted constructs related to “surprisingness”. Surprisingness values are in the range between 0 (“completely expected”) and 1 (“completely unexpected”).

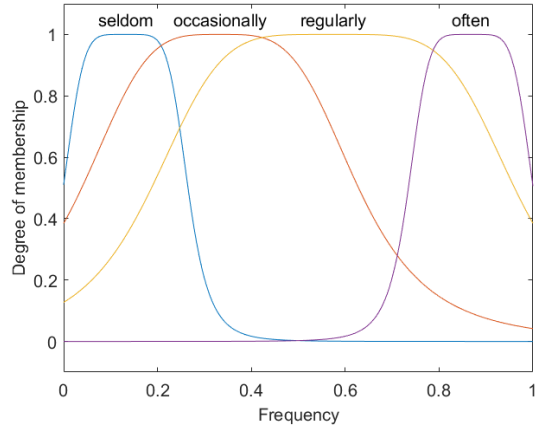


Figure 2: Depicted constructs related to event frequency. Frequency values are in the range between 0 (“never”) and 1 (“as often as it can be”).

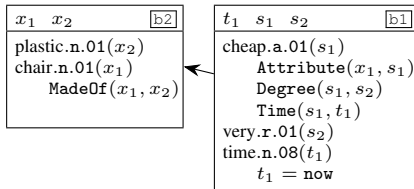


Figure 3: Example 81/2996. “This plastic chair is very cheap”.

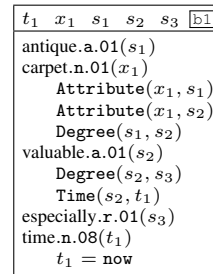


Figure 4: Example 20/0810. “Antique carpets are especially valuable”.

### 3.2 Semantic parsing

Semantic parsing attempts to map natural language to meaning representations. DRS parsing is a type of semantic parsing that targets Discourse Representation Structures (Kamp and Reyle, 2013), consisting of discourse referents and discourse conditions optionally organized into scopes (see Abzianidze et al. (2019) for an introduction to the scoped DRS representation).

Consider examples 81/2996 and 20/0810 from the Parallel Meaning Bank (Abzianidze et al., 2017), see figs. 3 and 4. While this representation is semantically very rich, it has limitations when it comes to the reasoning it may facilitate. For example, while there is clearly a relation between the meaning of the constructs “very cheap” and “especially valuable” (in terms of “price”), it is not easy to explore this relation. Indeed, the meaning of the words “cheap” and “valuable” is only defined by references to WordNet meanings, and while it is specified that both “very” and “especially” are degree modifiers, the effect of these modifiers is also only defined by references to WordNet.

One of the goals of our approach is attempting to be able to look “inside” the concepts, allowing to enable such reasoning. In this example this would mean relating both “cheap” and “valuable” to “price” using fuzzy meaning representations and defining “very” and “especially” as modifiers that transform those representations. In this way, the resulting DRS structures may express the meanings of the constructs in terms of the same underlying properties, facilitating additional types of reasoning.

## 4 Approach

The presented approach and the framework are largely based on the idea of exploring what various language constructs (e.g. “often”) tell us about possible values of different properties (e.g. “frequency”). We don’t know about a general way of deriving such knowledge from texts, however, see e.g. Runkler (2016). For this reason, our approach is dependent on having a predefined lexicon containing rather

rich semantic information. Currently, the lexicon is defined manually. While we are planning to work on obtaining some parts of this lexicon in an automated way and making the framework work with a broader lexicon, we consider it to be a different research topic. Our main interest is studying whether such information may improve understanding and reasoning capabilities of natural language understanding systems in general and semantic parsing systems in particular.

Relying on a rich semantic lexicon has other important implications on the design of the system. We are trying to make use of this information in the best possible way, as it may help to guide all the tasks that are part of the semantic parsing (e.g. syntactic parsing, handling multi-word expressions, choosing relevant word meaning, etc.). Also, we believe that any decision made during the natural language understanding process may need to be reversed as a result of the information obtained later in the process. For example, first we may parse “a few” as a multi-word expression, but later we may have to reconsider it, because we cannot arrive at any satisfactory interpretations that are based on that decision. For these reasons, rather than viewing the system as a pipeline that makes use of standard components working with different tasks, we design it as a single process that may take input from other specialized systems. Currently, the framework does not yet integrate with other systems, handling main tasks of the semantic parsing internally.

Given a lexicon, the framework parses natural language sentences to a variant of scoped DRS representation, loosely following Abzianidze et al. (2019). We extend the DRS to represent fuzziness by using *regions*, see also Kapustin and Kapustin (2015). Regions define which values of a certain property (e.g. “age”, “surprisingness”, “frequency”) are compatible with a certain word or utterance by the means of either fuzzy sets or compatibility intervals. Composition of meaning in a sentence is modeled by the application of word-operators to words-operands, and whether a word-operator may accept a particular word as an operand is based on various syntactic and semantic tags. The framework processes words in the order they appear in the input sentence, performing a series of transitions corresponding to the decisions made during the semantic parsing. As several alternative transitions are often possible, transitions form a search tree. Paths in this tree correspond to different interpretations of the sentence, and each parse is assigned a score. Finally, to find the best interpretation of the sentence, we traverse the tree, taking the scores into account.

The framework is written in Haskell and is under active development. While it is not open source software at the moment, we encourage everyone interested in the framework to contact the authors.

In this paper we describe our approach in general, and the framework in its current state. First, we briefly describe the main components of the framework, and then discuss the role that fuzzy meaning representations play in the framework.

#### 4.1 Operator application

We model composition of meaning in a sentence by application of word-operators to words-operands, somewhat similar to predicate-argument structure. Such applications give rise to new *meaning elements* (meaning composition units within a sentence). Whether a word-operator may accept another word as an operand is based on various syntactic and semantic tags, see also Abzianidze and Bos (2017). The tags may be defined in the lexicon, and in some cases calculated (for example, modeling transfer of some properties from an individual word to a compound). In the future, we plan to add support for tag expression levels, capability of inferring of tags (for example, in case of coercion), and also using both external sources and machine learning techniques to obtain the tags for the words that are not in the lexicon.

We use the following data type to represent a particular state in the semantic parsing that we call *application data*. This data type contains:

- The list of all *active operators* (word-operators awaiting their arguments).
- *Available operands* (seen words and meaning elements not yet assigned to any active operator).
- New meaning elements (created as a result of previous applications).
- Unprocessed sentence words.

- Current DRS.
- Current score of the interpretation.

Consider a somewhat shortened example, representing a state in the semantic parsing.

```

1 Application data:
2   ActiveOperators: [
3     Definition:
4       Description: after,
5       Operands: [
6         Name: ReferenceTime, Needed tags: [TimeMoment], Core: False},
7         Name: TimeInterval, Needed tags: [TimeInterval], Core: False}
8       ],
9     MatchCriteria:
10      OneOf: [ReferenceTime, TimeInterval],
11      Location: (5, 5),
12      OperandMatches: [
13        Definition:
14          Name: TimeInterval,
15          Needed tags: [TimeInterval],
16          Core: False
17        MatchData:
18          Description: just (a few (minutes)),
19          Tags: [Quantity, TimeInterval],
20          Location: (1, 4),
21          DrsReferentName: x1
22      ],
23      AvailableOperands: [],
24      MeaningElements: [
25        Description: starting,
26        Tags: [TimeMoment],
27        Location: (6, 6),
28        DrsReferentName: x2
29      ],
30      RemainingWords: []

```

According to the example, there is one active operator “after”. The operator’s definition (lines 3-10) says that it may accept two operands referred to as “ReferenceTime” and “TimeInterval”, and that both of them are optional. To be accepted as the operands, the candidate words or meaning elements need to have tags “TimeMoment” and “TimeInterval”, respectively. The operator is considered sufficiently matched if at least one of these operands is matched (line 10).

Lines 12-22 say that one of these operands, “TimeInterval”, is currently matched with the meaning element corresponding to the construct “just a few minutes”. This meaning element has tags “Quantity” and “TimeInterval”, originates from the words 1-4 in the sentence and corresponds to the discourse referent  $x_1$  in the output DRS.

Lines 24-29 say that the next meaning element to consider is “starting”. It has tag “TimeMoment”, originates from word 6 in the sentence and corresponds to the discourse referent  $x_2$  in the output DRS. Based on the tags, this meaning element can be matched with the “ReferenceTime” operand of the operator “after”.

## 4.2 DRS

Ultimately, all operator applications result in the modification of the output DRS. Consider the result of the semantic parsing of the construct “just a few minutes after starting”.

```

1   Ref x1 <- just (a few (minutes))
2   Quantity (x1)
3   Object (x1, "minutes")
4   Value (x1, "Countable": [0.10-0.20 -- 0.50-0.70])
5   Perceived (x1, "Perceived quantity": [0.10-0.20 -- 0.30-0.40])
6   Ref x2 <- starting
7   TimeMoment (x2)
8   Ref x3 <- starting
9   Process (x3)
10  StartTime (x3, x2)

```

```

11 Ref x4 <- after (starting, just (a few (minutes)))
12 TimeMoment (x4)
13 ReferenceTime (x4, x2)
14 Value (x4, "Time": [0.00-0.10 -- 0.50-0.80])

```

Lines 1-5 define a discourse referent  $x_1$  corresponding to the result of applying operator “just” to the meaning element “a few minutes”, along with several discourse conditions related to that referent. Predicate “Quantity” defines the type of the referent  $x_1$ . Relation “Object” defines the unit of measure which is “minutes”. Relation “Value” defines the value of the quantity, given by the compatibility interval  $[0.10-0.20 \text{ -- } 0.50-0.70]$  over the property “countable”. Relation “Perceived value” defines the quantity’s perceived value, given by the compatibility interval  $[0.10-0.20 \text{ -- } 0.30-0.40]$  over the property “perceived quantity”. Both of the compatibility intervals come from the definitions of the operators for the words “a few” and “just”.

Lines 6-10 define discourse referents  $x_2$  and  $x_3$ , both created during the *standalone application* (application without arguments) of the operator “starting”.  $x_2$  corresponds to the process, and  $x_3$  — to the starting moment of the process.

Lines 11-14 define discourse referent  $x_4$ , corresponding to the time moment defined by the operator “after”, applied to “just a few minutes” and “starting”. This time moment is defined with regards to the time moment  $x_2$ , and its value is given by the time interval  $[0-0.1 \text{ -- } 0.5-0.8]$  that comes from the definition of the operator for the word “after”.

### 4.3 Tree search

As previously mentioned, we model the task of finding the best interpretation of the sentence as a tree search task. Each branch in the tree corresponds to a certain decision in the semantic parsing process, from more syntactic (e.g. treating two words as a multi-word expression to more semantic (exploring a certain meaning of a word, applying an operator to arguments, etc.). Each search path in the tree corresponds to a potential interpretation of the sentence, and all the interpretations are scored. Some examples of the scores include:

- Ratio between the number of the matched operands and the numbers of the operands supported by the operator (optionally considering their importance).
- Distance between the operator and the operands in the sentence (reflecting that placement of various modifiers and function words relative to the word they specify varies between languages, and how strict the placement needs to be followed also depends on the words and the language).
- Different heuristics (see section 4.5).

Various types of scores are currently under development.

In addition to the automatic search mode, the framework supports an interactive mode where the user may choose which search path to explore next. In this mode, all paths in the tree are written to files that are updated as the search proceeds. The files contain various information about the search paths, in particular its status, scores, current output DRS, current application data and the steps the path consists of. Consider a somewhat shortened example of a search path to the semantic parsing of the construct “just a few minutes after starting”.

```

[ProcessLexeme("just"), ProcessMeaningElement("just"), NewActiveOperator("just"),
ProcessLexeme("a few"), ProcessMeaningElement("a few"), NewActiveOperator("a
few"), ProcessLexeme("minutes"), ProcessMeaningElement("minutes"), OperandMatches
("a few":"Countable"->"minutes"), Apply("a few"), OperandMatches("just":
["Quantity"->"a few (minutes)"]), Apply("just"), NewAvailableOperand("just(a few
(minutes))"), ProcessLexeme("after"), ProcessMeaningElement("after"),
AvailableOperandAllocation("after":["TimeInterval"->"just (a few (minutes))"]),
OperandMatches("after":["TimeInterval"->"just (a few (minutes))"]), ProcessLexeme(
"starting"), ProcessMeaningElement("starting"), ApplyStandalone("starting"),
OperandMatches("after":["ReferenceTime"->"starting"]), Apply("after"),
NewAvailableOperand("after(starting, just(a few (minutes)))") ]

```

This shows how the semantic parsing is separated into smaller steps, from parsing new lexemes to matching operands and applying operators. All these steps form a path in a single search tree.

## 4.4 Lexicon

The lexicon is used to define all the words with special meaning for the framework, i.e. operators. While the system may handle words not present in the lexicon, this is currently quite limited (they are treated as strings). At the moment, all the semantic tags also have to be defined in the lexicon, and the words not present in the lexicon are treated as if they didn't have any tags. However, we plan to use external sources and machine learning techniques to obtain semantic tags for the words that are not in the lexicon.

Currently, the lexicon is described using an embedded domain specific language in Haskell, but we plan to use a configuration language in the future.

For example, consider this piece of configuration:

```
oneWord "just"
  & addOperands [quantityOperand]
  & addNewCondition
    (operandRef Operands.quantity)
    ( relation
      Roles.perceived
      (regionValue (IntervalRegion Properties.quantity Intervals.low)))
  & setResultReferent (operandRef Operands.quantity)
  & addTagAction (AddTagsFrom Operands.quantity)
```

This says the following:

- “just” is a single-word lexeme.
- In this meaning, it takes one mandatory operand that needs to have a tag “Quantity”.
- When applied, “just” will add a new discourse condition “Perceived” to the discourse referent associated with this operand (here referred to as  $x_1$ ). The value of this relation is set to the compatibility interval “Intervals.low” (e.g.  $[0.1-0.2 \text{ -- } 0.3-0.4]$ ).
- The result of the application of “just” to an operand gives rise to a new meaning element that will have the same tags as the argument.
- The resulting meaning element is associated with the discourse referent  $x_1$ .

## 4.5 The role of fuzzy meaning representations

Using fuzzy meaning representations, the framework allows us to express the meaning of words in terms of the same underlying properties. For example, “seldom”, “occasionally”, “regularly”, “often” can all be related to frequency. This allows the system, in particular, to compare meanings of sentences containing related words.

The framework allows us to model the meaning of compound constructs in a compositional way, using operator application. In case of modifiers this means transformation of the original fuzzy set or compatibility interval by the modifier, and the result is again either a fuzzy set or a compatibility interval over the same underlying property. This means that the system can also compare meanings of sentences containing compound constructs, for example, “after”, “soon after”, “slightly after”, “right after”, “five minutes after”.

Expressing meaning of the constructs in terms of underlying properties also means that the system can in some cases decompose the meaning of the constructs in several parts. Let's discuss several examples.

- (1) Just a few minutes after starting, it has already completed cleaning of the whole room.
- (2) Only a few minutes after starting, it has already completed cleaning of the whole room.
- (3) ? Just an hour after the movie started, he realized it was a different title.
- (4) Only an hour after the movie started, he realized it was a different title.

Consider (1) and (2). While the words “a few” and “minutes” tell us about the actual duration of the process, the meaning of the words “just” and “only” in this context is different. Rather than describing the actual duration, they tell us how the duration is perceived (in this case “shorter, than expected”). This is also related to mirativity, or surprise, see e.g. Zeevat (2013), Zeevat (2009). In our approach,



semantic parsing of “just a few minutes” results in two fuzzy sets or compatibility intervals, one defined over property “quantity” and one defined over property “perceived quantity”. This allows the systems using these representations to analyze and compare the meanings of constructs like “just a few minutes” and “a couple of minutes”) property by property.

It is interesting to note that “only” in a similar context may actually have opposite meanings, “shorter than expected” (2) and “longer than expected” (4). However, “just” in a similar context always means “less than expected”, or “shorter than expected”, and this essentially makes (3) either ironic or infelicitous. If a system has information about both communicated perceived and actual values, it may detect irony in case of their disagreement.

While detection of irony in practice would often require some extra domain-specific knowledge e.g. in (3) it is necessary to know that “one hour” is a long time in this domain, often analysis of fuzzy meaning representations defined over the same or related properties may help to assess text understanding level and detect contradictions, tautologies and other situations that may suggest an infelicitous sentence (or a misunderstanding of a sentence by the system). Let’s discuss several examples.

- (5) ? This car is slow and very quick.
- (6) ? He is young but not too old.
- (7) ? The pizza is slightly medium.
- (8) ? I go there very always.
- (9) ? I will be there approximately in the evening.
- (10) Only a few people like sushi.
- (11) ? Only a great number of people like sushi.

Consider (5). Suppose “slow” and “quick” are defined by the compatibility intervals  $[0 \text{ -- } 0.2-0.3]$  and  $[0.7-0.8 \text{ -- } 1]$  over the property “speed”. Their intersection is empty, indicating a contradiction as long as the use of “and” suggests that both “slow” and “quick” should be possible at the same time.

Consider (6). Suppose “young” and “old” are defined by the compatibility intervals  $[0-18 \text{ -- } 30-50]$  and  $[70-80 \text{ -- } 100]$  over the property “age” (see also section 3.1). Modifier “too” would shift “old” even more to the right edge of the scale. However, the use of “but not” assumes some non-empty intersection between “young” and “too old”, which is not the case.

Consider (7). Suppose “medium” is defined by the compatibility interval  $[0.3-0.4 \text{ -- } 0.6-0.7]$  over the property “size”. Modifier “slightly” would normally shift the compatibility interval to the middle of the scale, making the construct less polar. However, the compatibility interval is already in the middle, and the shift has no effect.

Consider (8). Suppose “always” is defined by the compatibility interval  $[0.95-0.99 \text{ -- } 1]$  over the property “frequency”. Modifier “very” would normally shift the compatibility interval to the edge of the scale, making the construct more polar. However, the compatibility interval is already at the very edge, and the shift has virtually no effect.

Consider (9). Suppose “evening” is defined by the compatibility interval  $[0.5-0.6 \text{ -- } 1]$  over the property “time” where “0.5” means 6PM and “1” means 12AM. Modifier “approximately” would normally expect a rather narrow compatibility interval and make it wider. However, the compatibility interval is already rather wide.

While (10), even if not true, is a valid sentence, (11), on the other hand, seems infelicitous, if not ungrammatical. Suppose that “only” is defined by the compatibility interval  $[0-0.2 \text{ -- } 0.3-0.4]$  over the property “perceived quantity”, and “a great number” is defined by the compatibility interval  $[0.7-0.8 \text{ -- } 1]$  over the property “quantity”. The use of “only” tells us that the quantity is perceived as small and presupposes that the quantity is indeed small, so the compatibility intervals should agree, but they do not.

With use of fuzzy meaning representations, many of these situations may be detected computationally, possibly meaning that the phrase is infelicitous, or, more likely, that the system’s interpretation of the phrase is incorrect. We are currently working on development of similar heuristics in the framework,

partly drawing upon some of the ideas from Kapustin and Kapustin (2015), and they are going to take part in the scoring of the interpretations.

## 5 Discussion

One of the primary design goals for the approach and the framework is extensibility, as only a little part of the lexicon can be defined manually. Short-term, we are looking at using external sources and machine learning techniques to obtain semantic tags for the words not present in the lexicon. Long-term, we are planning to use various knowledge bases, synonym dictionaries and machine learning techniques to let the system learn more words.

At the same time, we would like to emphasize that our main research goal is exploring how natural language understanding systems in general and semantic parsers in particular may benefit from using rich quantitative meaning representations, focusing somewhat less on obtaining these representations automatically at this point. We know how to automatically learn some fuzzy sets, see e.g. Runkler (2016), and maybe we can find out how to learn more. We would like to study whether we should, and what understanding and reasoning capabilities this may facilitate, compared to what is supported by easily learnable but not easily linguistically interpretable representations like word embeddings.

As a reviewer pointed out, it is also interesting to compare fuzzy meaning representations to Universal Decompositional Semantics (White et al., 2016) in terms of the aspects of meaning they can express. Universal Decompositional Semantics is another quantitative representation and annotation scheme that uses continuous scales and is based on linguistic theory. While it is very different from fuzzy sets and compatibility intervals, we think it is worthwhile to explore how these representations could be used together to model different aspects of semantics.

We believe that fuzzy meaning representations may improve capabilities of the systems working with natural language as they provide linguistically interpretable projections (Kapustin and Kapustin, 2019b) of the construct meanings onto certain properties. These may allow natural language understanding systems to interpret meanings of the words in a quantitative way, allowing to analyze and compare them with higher “precision”. For example, instead of just knowing that the words are similar, the system may analyze in details how they are similar or different, e.g. which word is wider, which one is more polar, which situations can be described by both words and which — only by one of them.

Seeing whether natural language understanding systems can actually use some of the described ideas and techniques, and being able to evaluate this in practice will require more research, and we hope that our approach and the software framework can be a useful contribution to the study of this complex field.

## Acknowledgements

We thank Csaba Veres, Vadim Kimmelman, Rik van Noord, Costanza Marini and anonymous reviewers for helpful discussions, comments and feedback. We thank Rik van Noord and Lasha Abzianidze for their assistance with preparation of the illustrations for the paper.

## References

- Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.
- Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.
- Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247.
- Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden, May. Association for Computational Linguistics.

- Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.
- Claire Bonial, William Corvey, Martha Palmer, Volha V Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489. IEEE.
- Joao P Carvalho, Fernando Batista, and Luisa Coheur. 2012. A critical survey on the use of fuzzy sets in speech and natural language processing. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–8. IEEE.
- Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2-3):281–332.
- Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press, Cambridge, Ma., USA.
- Harry M Hersh and Alfonso Caramazza. 1976. A fuzzy set approach to modifiers and vagueness in natural language. *Journal of Experimental Psychology: General*, 105(3):254.
- Hans Kamp and Uwe Reyle. 2013. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.
- Michael Kapustin and Pavlo Kapustin. 2015. Modeling meaning: computational interpreting and understanding of natural language fragments. *arXiv preprint arXiv:1505.08149*.
- Pavlo Kapustin and Michael Kapustin. 2019a. Modeling language constructs with compatibility intervals. In *Proceedings of the IWCS 2019 Workshop on Computing Semantics with Types, Frames and Related Structures*, pages 49–54, Gothenburg, Sweden, June. Association for Computational Linguistics.
- Pavlo Kapustin and Michael Kapustin. 2019b. Modeling language constructs with fuzzy sets: some approaches, examples and interpretations. In *Proceedings of the 13th International Conference on Computational Semantics - Student Papers*, pages 24–33, Gothenburg, Sweden, May. Association for Computational Linguistics.
- Pavlo Kapustin and Michael Kapustin. 2020. Language constructs as compatibility intervals: a small-scale experiment. In *ExLing 2020: Proceedings of the 11th International Conference of Experimental Linguistics*. International Society of Experimental Linguistics. Accepted for publication.
- Pavlo Kapustin. 2015. Computational comprehension of spatial directions expressed in natural language. Master’s thesis, The University of Bergen.
- Vilém Novák. 2008. A comprehensive theory of trichotomous evaluative linguistic expressions. *Fuzzy Sets and Systems*, 159(22):2939–2969.
- Vilém Novák. 2017. Fuzzy logic in natural language processing. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, pages 1–6. IEEE.
- Vilém Novák. 2020. The concept of linguistic variable revisited. In *Recent Developments in Fuzzy Logic and Fuzzy Sets*, pages 105–118. Springer.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926.
- Thomas A Runkler. 2016. Generation of linguistic membership functions from word vectors. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*, pages 993–999. IEEE.
- Mark Steedman. 2000. *The syntactic process*, volume 24. MIT press Cambridge, MA.

- Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.
- Lotfi A Zadeh. 1971. Quantitative fuzzy semantics. *Information Sciences*, 3(2):159–176.
- Lotfi A Zadeh. 1972. A fuzzy-set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics*.
- Lotfi Asker Zadeh. 1975. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199–249.
- Lotfi Asker Zadeh. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28.
- Henk Zeevat. 2009. “only” as a mirative particle.
- Henk Zeevat. 2013. Expressing surprise by particles. In *Beyond Expressives: Explorations in Use-Conditional Meaning*, pages 297–320. Brill.