# Logic-based machine comprehension for chatbots

Sergiu Redeca and Adrian Groza
Technical University of Cluj-Napoca,
sergiuredeca@gmail.com, Adrian.Groza@cs.utcluj.ro

***Abstract*—We target here the interpretation of natural language by means of Equational First Order Logic (FOL). The communicative acts of the human agent are automatically converted into a First Order Logic theory. The models of the FOL theory are analysed by Mace4 model finder. To goal of the chatbot is to reduce the number of interpretation models. With a single interpretation model, the chatbot is sure about the human agent preferences. To reduce the number of models, the agent asks questions, while the corresponding answers are used to restrict the models. The order in which these questions are asked is extremely relevant to effectevely reduce the huge number of interpretation models of a FOL theory. We apply here an entropy-based method to improve the dialogues between a conversational agent and a human agen. The new method computes the entropy on the set of possible interpretation models. The experiments indicated that with entropy-based chatbots it is easier to estimate the budget of questions needed to elicit client preferences.**

***Index Terms*—machine comprehension, discourse understanding, chatbot, interpretation models, first order logic**

## I. Interpretation models in FOL

Here is a puzzle for you: *Assume the text "Romeo and Julieta are in love" is automatically translated by a machine learning-based translator into "Romeo is in love" and "Julieta is in love". This is formalised in FOL with: $A_1$: $\exists x,\ love(romeo, x)$ and $A_2$: $\exists x,\ love(julieta, x)$. How many interpretation models of this first order logic theory are there?*

The human agent has a single interpretation model: there are two individuals, Romeo ($r$) and Julieta ($j$), that love each other (see Fig. 1). Here, $r$ stands for $romeo$, $j$ for $julieta$,

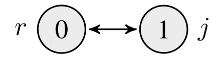


Fig. 1: The unique interpretation model of the human agent.

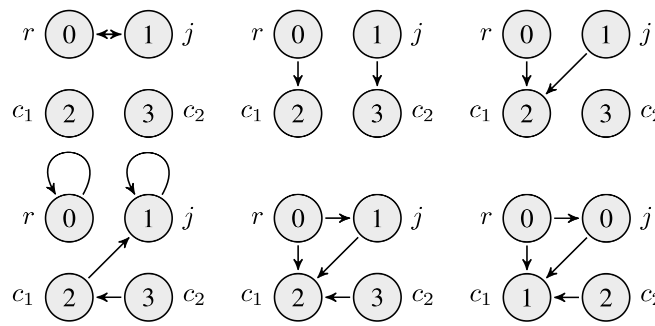


Fig. 2: Sample of interpretation models for the software agent.

Instead, the logical agent has a huge number of interpretation models, even for tiny theories like this. To answer this question we used the Mace4 model finder [1]. Mace4 is able to compute interpretation models of first order logic theories with finite domains. Here, we close the domain to four individuals (i.e. we assume there are only four objects in the domain). Axiom $A_1$ says that $romeo$ loves an individual $x$, while axiom $A_2$ says that $julieta$ loves an individual $x$, that is not necessarily the same, since each variable has its own existential quantifier. Mace4 solves the puzzle for you: there are 278,528 models.

Since this number of models might been unexpected (recall that the domain was restricted to four individuals only), we analysed the generated models. A sample of these models is illustrated in Fig. 2. Here $c_1$ and $c_2$ are the Skolem constants generated for the existential quantifiers in $A_1$ and $A_2$. As the domain is closed to four individuals we work only with the set of integers $\{0, 1, 2, 3\}$. The first model (first row, left) is consistent with the human interpretation: $romeo$ and $julieta$ do love each other. Note also that all four individuals are distinct: $r \rightarrow 0$, $j \rightarrow 1$, $c_1 \rightarrow 2$, $c_2 \rightarrow 3$. In the second model (first row, center), $romeo$ loves an individual $c_1$, while $julieta$ loves a distinct individual $c_2$. In the third model (second row, left), both $romeo$ and $julieta$ love the same individual $c_1$. Moreover, no one sad that the $love$ relation is not reflexive. One such model is the fourth one (second row, right), where both $romeo$ and $julieta$ love each other. The variety of the models is also increased by different possible love relations involving $c_1$ and $c_2$. For instance in the fourth model, $c_1$ loves $heloise$ and $c_2$ loves $c_1$. Similarly, no one sad that someone can love only one person at the same time. Therefore, the fifth model (third row, left) is possible. Here, $romeo$ loves both $julieta$ and $c_1$. The largest influence is given by the fact that the logical agent can interpret that some individuals are not distinct. In the sixth model (third row, right), $romeo$ and $julieta$ are interpreted by Mace4 as the same individual ($r \rightarrow 0$, $j \rightarrow 0$) referred by two distinct names.

The above analysis explains how the logical agent has indeed 278,528 interpretation models for the simple sentence *Romeo and Julieta are in love*. Moreover, all these models are equally plausible for the software agent. Given this gap (278,528 models vs. one model of the human agent), the natural question is *How the two agents understand each other?*

The task is to reduce the number of interpretation models for the software agent. We consider dialogues between a logical agent (i.e. a chatbot) and the human agent in the context of customer service [2]. We illustrate the proposed conceptual and technical instrumentation on the pizza-ordering scenario.

## II. Formalising the ordering pizza scenario

The pizza seller chatbot aims to support clients choosing a pizza according to some available ingredients. There are two technical objectives: (1) reducing the interpretation models to a single one; and (2) empowering the chatbot with a queering strategy with the smallest number of questions to reach the single interpretation model.

We formalise in FOL the following domain knowledge. Assume that pizza is the only product for sale: $\forall x, x = Pizza \leftrightarrow product(x)$. The available ingredients are sauce, fish, or cheese. $\forall x\ ingredient(x) \leftrightarrow\ x = Sauce\ \vee x = Fish\ \vee x = Cheese$. Let also three locations for the ingredients: $\forall x\ location(x) \leftrightarrow\ x = AtlanticOcean \vee x = BlackSea \vee x = PacificOcean$. Products, ingredients and locations are disjoint: $\forall x\ product(x) \rightarrow \neg ingredient(x) \wedge \neg location(x)$, $\forall x\ ingredient(x) \rightarrow \neg location(x) \wedge \neg product(x)$, $\forall x\ location(x) \rightarrow \neg ingredient(x) \wedge \neg product(x)$.

Next, we introduce the binary relation "`from`", between an ingredient and a location: $\forall x\ forally\ from(x, y) \rightarrow ingredient(x) \wedge location(y)$. Assume that $fish$ is the unique ingredient from the Atlantic Ocean: $\forall x\ from(x, AtlanticOcean) \rightarrow x = Fish$. Similarly, $fish$ is the single ingredient from the Black Sea: $\forall x\ from(x, BlackSea) \rightarrow x = Fish$. Next we specify that there are no available ingredients from the Pacific Ocean: $\neg\exists x\ from(x, PacificOcean)$. Hence, fish origins only from the locations: $\forall x\ from(Fish, x) \rightarrow x = BlackSea \vee x = AtlanticOcean$. The relation *contains* has domain a product and has range an ingredient: $\forall x \forall y\ contains(x, y) \rightarrow product(x) \wedge ingredient(y)$. Let two types of sauce: sweet or spicy. To represent the fact that the sauce can be of several kinds, we introduce the relation *varied*: $varied(Sauce)$. The relation is valid only for the types of sauce, not having several types of cheese or fish: $\forall x\ varied(x) \leftrightarrow x = Sauce$. We also formalise the fact that something that is in many ways is varied: $\forall x\ \forall y\ is(x, y) \rightarrow varied(x)$. Next, we state the only two types of sauce available: $\forall x\ is(Sauce, x) \rightarrow x = Spicy\ \vee\ x = Sweet$. The pizza domain can be extended with new locations (e.g. *Jiu River* and *Yellow Sea*), with new kinds of cheese (e.g. *white*, *yellow* and *blue cheese*), or other sauces (e.g. Taiwanese):

$$is(Sauce, Taiwanese) \leftrightarrow from(Fish, YellowSea) \quad (1)$$

$$is(Cheese, Blue) \rightarrow \neg from(Fish, Jiu) \quad (2)$$

$$is(Sauce, Spicy) \rightarrow \neg is(Cheese, Yellow) \quad (3)$$

$$is(Cheese, White) \rightarrow \neg is(Sauce, Sweet) \quad (4)$$

Our task is to match this FOL theory against the client preferences. Since these preferences are conveyed in natural language, we need to automatically translate into FOL.

## III. Converting natural language to FOL

To convert natural language sentences into FOL, we rely on the NLTK library. First, we use the classical pipeline of tokenisation, stop words removing, lemmatisation to obtain a list of words consisting of lowercase letters. Second, we use the NLTK library to retrieve the part of speech for each token. Starting from a list of words that are represented as strings: `['pizza','with','fish']` we get a list of word pairs and the part of speech associated with them: `[('pizza','NN'),('with','IN'),('fish','NN')]`. Third, we create a grammar based on the parts of speech. A grammar for *"pizza with fish"* can be used to identify a structure such as: {<NN.*><IN><NN.*>}. This expression identifies structures that are made up of a noun, followed by a preposition, followed by another noun.

Parsing rules are also defined for processing sentences such as *"The sauce should be spicy."*: <NN.*><.*><VB><JJ> The expression identifies sentences containing a noun, followed by a verb and an adjective. We consider that the user will often express his intention in the form of sentences that contain structures such as *"pizza with fish"*, for example. Similar intentions may contain a negation that will slightly change the user's intention. Let for instance the sentence *"pizza with no cheese"*, handled by the following expression: {<NN.*><IN><DT><NN.*>}. The expression matches sentences that start with a noun, followed by a preposition, followed by a negation and that end with a noun.

In the previous case, the negation applies to the ingredient put on *pizza* (e.g. *cheese*). The negation is addressed to the second noun. However, the case occurs if the negation is addressed to the first noun (i.e.*pizza*). For this, the agent needs to identify a structure similar to *"no pizza with cheese"*: {<DT><NN.*><IN><NN.*>}. The expression identifies structures that begin with a negation, followed by a noun, followed by a sentence, and that end with another noun.

Another less common case is that in which two negations are encountered: one on the first noun, and the other on the second, e.g. *"I want no pizza with no cheese."*. The structure *"no pizza with no cheese"* is handled by {<DT><NN.*><IN><DT><NN.*>}. The expression identifies structures that begin with a negation, followed by a noun, then a preposition, which is followed by a negation and a noun.

Another pattern is given by sentences like *"I want a pizza with spicy sauce."*. Upon identifying the intention to have a pizza with sauce, it must also be spicy: {<NN.*><IN><NN.*><NN.*>} and {<NN.*><IN><JJ><NN.*>}. Such expressions capture structures consisting of a noun followed by a preposition, which is followed by another noun or adjective, depending on the ability of the NLTK library to correctly identify the part of speech, and which ends with a noun. The grammar created is as follows:

```
CASE1:{<NN.*><IN><NN.*>}
      {<NN.*><.*><VB><JJ>}
CASE2:{<NN.*><IN><DT><NN.*>}
CASE3:{<DT><NN.*><IN><NN.*>}
CASE4:{<DT><NN.*><IN><DT><NN.*>}
CASE5:{<NN.*><IN><NN.*><NN.*>}
      {<NN.*><IN><JJ><NN.*>}
```

Fourth, we identify the logical relationships received from the user. We add the elements in the domain to which the user refers, the context that the agent has following the dialogue with the user. To find the known relations we added in the logical formulas the relations $x_1$ and $x_2$:

$$\forall x \; x_1(x) \leftrightarrow product(x) \vee ingredient(x) \quad (5)$$
$$\vee location(x) \vee varied(x)$$
$$\forall x \forall y \; x_2(x,y) \leftrightarrow is(x,y) \vee contains(x,y) \vee from(x,y) \quad (6)$$

The formula 6 is a true relation for any element in the domain with the property that one of the relations with arity 1 (i.e. (*product*, *ingredient*, *location*, *varied*) is true for that element. That is, the relation is true when the element in the domain is a product, an ingredient, a location, or if it can be of several types (such as sauce can be sweet or spicy). The formula 6 is similar to the formula 6, but it is valid for relations with arity 2 (*is*, *contains*, *from*). The relationship is true if the second element is a property of the first element, or if the first element in the domain contains the second, or if the first element in the domain has its origin in the element on the second position. The formulas 6 and 6 therefore have an auxiliary role, and are used to identify possible relationships that may occur between user-transmitted syntax elements, and initially identified using our grammar. The data received from the user is therefore interpreted to identify the possible relationships $x_1$ and $x_2$. The relationships will then be taken into account when searching for models with Mace4.

## IV. Reducing interpretation models through questions

With both the pizza domain knowledge and knowledge from the dialogue formalised in FOL, Mace4 has the role to identify all the interpretation models.

Given only the pizza domain, without the extensions from the 1-4 equations, there are 48 models. As dialogue moves are analysed and formalised these models are reduced or expanded. The task of the chatbot is to select questions whose answers reduce the number of models, hence the epistemic uncertainty about the current order.

A step by step example will illustrate how the logical agents builds its FOL theory and interpretations. Let the client starting the dialogue with `I want a pizza with fish`. The new relationship added in to the FOL theory is $x_2(Pizza, Fish)$. The answer matches CASE1 in the grammar. Based on this, the objects *pizza* and *fish* are extracted and included in the domain.

The agent must now identify the relationship that can be established between the two elements in the field. The three relationships that satisfy this property are *is*, *contains* and *from*. The only relation between the two elements that is true, according to the current FOL theory, is that of *contains*, but this will not be replaced directly in the field.

The number of possible models computed by Mace4 is 24 (see Table I). Ignoring these patterns, omitting the determination of the relationship would bring the greatest gain of information if it were clarified whether this relationship is true or false. However, such an approach is implemented in the case of the agent that uses entropy. Without entropy, the chatbot randomly chooses the relationship whose truth value needs to be verified $is(Sauce, Spicy)$. The corresponding question is generated in natural language: `Should the sauce be spicy?`

TABLE I: Reducing the number of models through dialogue (the case of the random agent)

| Step | FOL theory | Models |
|---|---|---|
| 0 | Pizza domain | 48 |
| 1 | $x_2(pizza, fish)$ | 24 |
| 2 | $\neg is(sauce, spicy)$ | 18 |
| 3 | $\neg from(Fish, AtlanticOcean)$ | 12 |
| 4 | $\neg x_2(Pizza, Sauce)$ | 4 |
| 5 | $contains(Pizza, Cheese)$ | 2 |
| 6 | $from(Fish, BlackSea)$ | 1 |

Assuming a *"no"* answer, the relation that sauce should not be spicy is added to the theory $\neg is(sauce, spicy)$. For the moment, the FOL theory has been extended with two relations: $x_2(Pizza, Fish)$, $\neg is(Sauce, Spicy)$. With these two pieces of knowledge, Mace4 computes 18 interpretation models. Based on these 18 models, the conversational agent proceeds by randomly determining which logical equation should be chosen in the current case. Let the selected relation: $from(Fish, AtlanticOcean)$. To obtain the truth value of this relation, the following question is displayed: `Should the fish be from the Atlantic?`

Assuming a negative answer, we have $\neg from(Fish, AtlanticOcean)$ Since there are 12 models, the chatbot continues the dialogue. The next considered relation $is(Sauce, Sweet)$ trigger the following question: `Should the sauce be sweet?`

Assume the following answer `I want pizza with no sauce`. The answer falls into the CASE2 of our grammar. The predicate derived from the user's statement is:$\neg x_2(Pizza, Sauce)$. Note that the predicates identified so far (step 4 in Table I) restrict the interpretations to four models only.

The next question chosen by the conversational agent corresponds to the $contains(Pizza, Cheese)$ relation: `Do you want pizza with cheese?` Given a positive answer, the chatbots adds the fact: $contains(Pizza, Cheese)$.

Mace4 outputs now two interpretation models. One of these models claims that the fish is from the Black Sea. The other claims that he is not from the Black Sea (recall that the client has already specified that he does not want fish from the Atlantic Ocean). Once again, the agent identifies the relation that introduces non determination, and makes the corresponding move: `Should the fish be from the Black Sea?` The positive answer "yes" triggers the fact: $from(Fish, BlackSea)$.

Since, there is a single interpretation model, the chatbot has successfully found the pizza that the client wants with six questions (six steps in Table I). Note that the grammar has accommodated two complex answers and four yes/no answers.

TABLE II: Reducing the number of models through dialogue (the case of an entropy based agent). The entropy is computed after the new relation is added in the logic domain.

| Step | Max entropy | FOL theory | Models |
|---|---|---|---|
| 0 | not used | Pizza domain | 48 |
| 1 | 1 | $x_2(Pizza, Fish)$ | 24 |
| 2 | 0.92 | $contains(Pizza, Cheese)$ | 12 |
| 3 | 1 | $\neg from(Fish, AtlanticOcean)$ | 8 |
| 4 | 0.81 | $from(Fish, BlackSea)$ | 4 |
| 5 | 0 | $\neg contains(Pizza, Sauce)$ | 1 |

TABLE III: Entropy after the first exchange of lines

| Relation | Entropy |
|---|---|
| $contains(Pizza, Cheese)$ | 1.00 |
| $from(Fish, BlackSea)$ | 0.92 |
| $from(Fish, AtlanticOcean)$ | 0.92 |
| $contains(Pizza, Sauce)$ | 0.81 |
| $is(Sauce, Sweet)$ | 0.81 |
| $is(Sauce, Spicy)$ | 0.81 |

## V. Using entropy on relations for shorter dialogues

We propose here a method of selecting questions based on entropy. The entropy can be used to find those predicates that lead to the greatest gain of information. The more information gain per answer, the shorter the dialogue. Thus, for each relation and for each combination of objects that can be associated using this relation, the goal is to identify the case in which the resulting entropy becomes minimal.

*Definition 1 (Entropy of a relation):* Given a set $\mathcal{M}$ of interpretation models, the entropy $e$ of the relation $r_i$ having the set of chosen objects $o \in O^{a(r_i)}$ is defined as:

$$e(\mathcal{M}, r_i, o) = -P(\mathcal{M}, r_i, o)\log_2(P(\mathcal{M}, r_i, o)) - (1 - P(\mathcal{M}, r_i, o))\log_2(1 - P(\mathcal{M}, r_i, o)) \quad (7)$$

where the probability of a relation $r_i$ to be true for a set of objects from the domain $o$ in a model $m \in \mathcal{M}$ is

$$P(\mathcal{M}, r_i, o) = \frac{\sum_{m \in \mathcal{M}} m(r_i, o)}{|\mathcal{M}|}, o \in O^{a(r_i)} \quad (8)$$

Here $a(r_i)$ represents the arity of the relation $r_i$, and $O$ is the set of all objects in the domain. $m$ is a function that represents the truth value of the relation $r_i$ for the set of objects $o$.

The relationship and the chosen objects in the domain are determined by the function $(r, o) = \text{argmax}_{(r_i,o)}\, e(\mathcal{M}, r_i, o)$, where $R$ represents all relationships in the domain, $r_i \in R$, $o \in O^{a(r_i)}$. Five questions are needed to identify the desired pizza: a pizza that contains fish, that is from the Black Sea, that contains cheese and sweet sauce, that is, pizza that is also desired in the example with the conversational agent based on random choice. After the first dialogue move `I want a pizza with fish`, the agent (i) computes the interpretation models, (ii) checks the relationships that differ between the possible models, (iii) determines which relationship is expected to give the most information gain.

TABLE IV: Entropy after second exchange of lines

| Relation | Entropy |
|---|---|
| $from(Fish, BlackSea)$ | 0.92 |
| $from(Fish, AtlanticOcean)$ | 0.92 |
| $contains(Pizza, Sauce)$ | 0.81 |
| $is(Sauce, Sweet)$ | 0.81 |
| $is(Sauce, Spicy)$ | 0.81 |

TABLE V: Entropy after the third exchange of lines.

| Relation | Entropy |
|---|---|
| $from(Fish, BlackSea)$ | 1.00 |
| $contains(Pizza, Sauce)$ | 0.81 |
| $is(Sauce, Sweet)$ | 0.81 |
| $is(Sauce, Spicy)$ | 0.81 |

In Table III, entropy is calculated using equation 8. There are 24 models: $|\mathcal{M}_1| = 24$. The $r_1 = from(Fish, BlackSea)$ relation is true in 8 models: $e(\mathcal{M}_1, from, (Fish, BlackSea)) = -\frac{8}{24} \cdot \log_2 \frac{8}{24} - (1 - \frac{8}{24}) \cdot \log_2(1 - \frac{8}{24}) = 0.92$.

After the first dialogue move, the relation with the largest entropy (see Table III is $contains(Pizza, Cheese)$ that triggers the corresponding question: `Do you want pizza with cheese?` Following a positive answer, there are 12 models remaining. The agent continues to analyse the models and computes the entropy for the relationships that differ. The relation chosen from the Table IV for verification is: $from(Fish, AtlanticOcean)$.

Similarly, the entropy computed for $from(Fish, BlackSea)$ using equation 8 is: $e(\mathcal{M}_2, from, (Fish, BlackSea)) = -\frac{4}{12} \cdot \log_2 \frac{4}{12} - (1 - \frac{4}{12}) \cdot \log_2(1 - \frac{4}{12}) = 0.92$. The question asked by the conversational agent is formulated according to this relationship: `Should the fish be from the Atlantic?`

Based on a negative answer, $\neg from(Fish, BlackSea)$ relation is added. The entropies calculated for the remaining relationships are presented in Table V. The relationship chosen for the following question is: $from(Fish, BlackSea)$. The generated question is: `Should the fish be from the Black Sea?`

Based on a positive answer, the fact $from(Fish, BlackSea)$ is added, resulting in four models. The remaining relations for which the entropy is calculated at this point are listed Table VI.

The relationship chosen based on entropy is $contains(Pizza, Sauce)$. Based on this is the conversational agent formulates and asks the question `Do you want`

TABLE VI: Entropy after the fourth exchange of lines

| Formula | Entropy |
|---|---|
| $contains(Pizza, Sauce)$ | 0.81 |
| $is(Sauce, Sweet)$ | 0.81 |
| $is(Sauce, Spicy)$ | 0.81 |

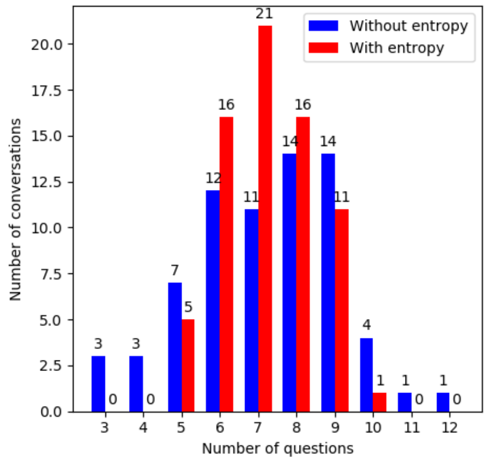


Fig. 3: Distribution of questions for both agents. The order is *fish from the Atlantic Ocean and sauce which is not spicy*.

`pizza with sauce?` Assuming a negative answer, Mace4 computes a single interpretation model.

## VI. Running experiments

We compare the entropy-based method to select questions against the random strategy. For each strategy, and for each scenario, 70 such dialogues were generated. The first statistical value calculated is the arithmetic mean of the questions needed to determine the necessary spices. We used the formula $average = \frac{total\ number\ of\ questions}{total\ number\ of\ conversations}$, where the total number of conversations is 70. The second statistical value that we consider is the *standard deviation* $\sigma$ of the number of questions needed to identify the exact order: $\sigma^2 = \sum_{i=1}^{N} x_i^2/N - \bar{x}^2$. The third value calculated is the minimum number and maximum number of dialogs used in a conversation. The results are in Table VII.

An example of distribution of questions appears in Figure 3. It represents the number of questions needed to determine the order. The user orders *fish from the Atlantic Ocean and sauce which is not spicy*. The extended logical domain is used. In this scenario, the means of the distributions for the two agents are about the same. The standard deviation is smaller for the agent using entropy. The agent with random choice of questions has the conversations both the minimum and maximum number of dialogues required. In another example presented in Figure 4, it can be seen that the mean of the agent which uses entropy requires on average a smaller number of questions for determining the pizza the user wants, but it can also require a bigger number of questions. In all the examples, the standard deviation is smaller for the agent using entropy.

Table VII contains statistical computations run scenarios for the extended logical domain. The minimum and maximum values of the question parameter required to determine the desired product, are represented with $m$ and $M$. The average number of dialogues used is represented with $a$ and the standard deviation is represented with $d$. The agent which uses random selection is represented with $R$ and the agent which uses entropy is represented with $E$. It can be observed that the agent using entropy obtains a lower standard deviation than the selection agent random selection of questions. Experiments suggest that when using the agent that uses entropy is easier to estimate the number of questions required for determining the product desired by the user.

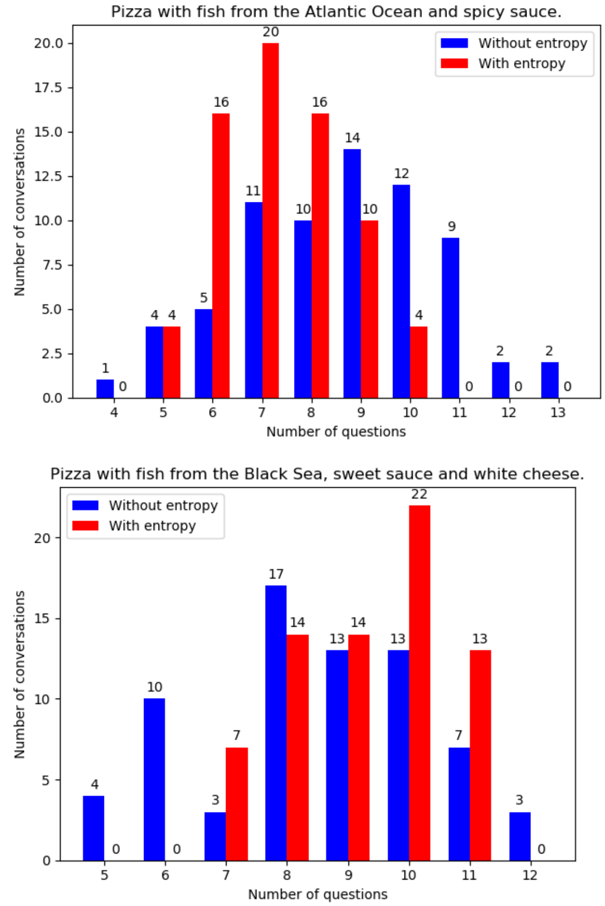


Fig. 4: A distribution of questions for both of the agents. Left: The user orders *pizza with fish from the Atlantic Ocean and spicy sauce*. Right: The user orders *pizza with fish from the Black Sea, sweet sauce and white cheese*.

## VII. Discussion and Related work

Most text inference methods have low semantic accuracy, but are highly robust. On the other hand, systems that rely on logics are accurate, but they are also fragile. In line with Garrete and Klein [3], our chatbot is based on NLTK toolkit [4] and Mace4 model finder.

Another method for structuring knowledge from natural language is *abstract meaning representations*. An abstract meaning representation (AMR) is a rooted in labeled graph designed to be easy-to-read for human agents, and easy-to-parse for software agents. The leaves of AMRs are labeled with concepts. The PropBank framesets are heavily used to abstract away the English syntax. A possible disadvantage could be that

TABLE VII: Statistical computations on running experiments

| # | Simulated order | $m_R$ | $m_E$ | $M_R$ | $M_E$ | $a_R$ | $a_E$ | $d_R$ | $d_E$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | fish from Jiu and spicy sauce | 5 | 5 | 12 | 10 | 7.94 | 7.4 | 1.73 | 1.22 |
| 2 | fish from Jiu and sauce which is not spicy | 3 | 5 | 12 | 10 | 6.95 | 7.65 | 2.08 | 1.25 |
| 3 | fish from the Atlantic Ocean and sauce which is not spicy | 3 | 5 | 12 | 10 | 7.22 | 7.21 | 1.92 | 1.20 |
| 4 | fish from the Atlantic Ocean and spicy sauce | 4 | 5 | 13 | 10 | 8.65 | 7.34 | 1.99 | 1.28 |
| 5 | fish from the Atlantic Ocean or the Black Sea and spicy sauce | 4 | 5 | 11 | 8 | 7.07 | 6.51 | 1.63 | 0.89 |
| 6 | fish from the Atlantic Ocean or the Black Sea and sweet sauce | 4 | 5 | 11 | 9 | 7.52 | 6.97 | 1.61 | 1.09 |
| 7 | fish from the Black Sea, sweet sauce and blue cheese | 4 | 7 | 13 | 10 | 8.12 | 8.5 | 1.84 | 0.99 |
| 8 | fish from the Black Sea, sweet sauce and white cheese | 5 | 7 | 12 | 10 | 8.52 | 9.28 | 1.83 | 1.25 |
| 9 | fish from Jiu, sweet sauce and white cheese | 5 | 7 | 12 | 11 | 7.97 | 9.01 | 1.48 | 1.18 |
| 10 | fish from Jiu or the Black sea, sweet sauce and white or blue cheese | 4 | 7 | 10 | 10 | 6.94 | 8.44 | 1.39 | 0.98 |
| 11 | fish from Jiu or the Atlantic Ocean, sweet sauce and blue cheese | 5 | 7 | 12 | 10 | 7.54 | 8.57 | 1.70 | 0.87 |
| 12 | fish from Jiu or the Atlantic Ocean, spicy sauce and blue cheese | 5 | 7 | 12 | 10 | 7.32 | 8.54 | 1.57 | 0.98 |
| 13 | fish from Jiu or the Atlantic Ocean, taiwaneese sauce and blue cheese | 3 | 9 | 12 | 12 | 7.10 | 10.10 | 2.05 | 1.17 |
| 14 | fish from Jiu or the Atlantic Ocean, taiwaneese sauce and yellow cheese | 3 | 7 | 11 | 10 | 6.04 | 7.91 | 1.85 | 0.93 |
| 15 | fish from Jiu or the Atlantic Ocean, taiwaneese sauce and white cheese | 4 | 7 | 12 | 10 | 7.55 | 8.01 | 1.71 | 0.76 |

AMRs are biased towards English. Another disadvantage is the lack of support for the inflectional morphology for tense and number. AMRs do not have the universal quantifier, so we cannot distinguish between real and hypothetical events [5]. Instead of AMR, we used here FOL theories. We relied on Mace4 to find the relation between the objects identified by the grammars.

*Discourse representation theory* also aims at dynamic interpretation of natural language. Each sentence is interpreted according to its influence over the discourse. Let the following example: *Bob saw a red bike in a show-window. He bought it.* The system should be able to conclude that *he* is referring to *Bob* and *it* is referring to the *bike*, which was *red*. Anaphoric pronouns can be represented as free variables linked to the corresponding antecedent variables [6] A similar problem we had was to identify the meaning of the yes/no answers from the user. For this, we used the NLP grammar rules.

There are several tools aiming to automatically translate natural language to some formal representation [7], [8], [9]. AllenNLP framework built on top of PyTorch is used for designing deep learning models [8]. The text is firstly represented as vector sequences. The sequences are then modified by being passed through a recurrent network to encode contextual information. Finally, the sequences are merged into a single vector using a recurrent network with averaging or pooling, or using a convolutional network. Instead of relying on deep learning, our agent rely mostly on symbolic computation. That is, instead of using the black box models learned from artificial neural networks, our agent looks into finite interpretation models generated by Mace4, that are white box.

## VIII. Conclusion

We deal here with interpretable models of natural language through first order logic. We propose a new method for reducing the number of questions required to clarify the intent of a natural language. The method uses the entropy of a relation, given a set of current interpretation models. The aim is quickly obtain a single interpretation model of dialogue.

Our experiments showed the agent using entropy obtains a smaller standard deviation than the agent with random selection of the questions. When using the agent that uses entropy, it is thus easier to estimate the budget of questions needed to elicit all preferences of client agents.

We are currently extending the the number of axioms in the pizza domain in order to run experiments on larger number of interpretation models. We are also working on generalising the entropy formulas for isomorphic models [10].

One can extend our work in several directions. First, since most bots follow pre-designed scripts, a possible extension would consider interleaving model reduction based on entropy with questions selected from scripts. Second, the seller bot can be augmented with argumentation and explanatory capabilities aiming at more trustful [11] conversations.

## References


[1] W. McCune, "Mace4 reference manual and guide," *arXiv preprint cs/0310055*, 2003.

[2] C. Chakrabarti and G. F. Luger, "Artificial conversations for customer service chatter bots: Architecture, algorithms, and evaluation metrics," *Expert Systems with Applications*, vol. 42, no. 20, pp. 6878 – 6897, 2015.

[3] D. Garrette and E. Klein, "An extensible toolkit for computational semantics," in *Proceedings of the Eight International Conference on Computational Semantics*, 2009, pp. 116–127.

[4] S. Bird and E. Loper, "NLTK: the natural language toolkit." Association for Computational Linguistics, 2004.

[5] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proc. of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013, pp. 178–186.

[6] J. van Eijck, "Discourse representation theory," *Encyclopedia of language and linguistics*, vol. 3, pp. 660–669, 2005.

[7] A. Groza and C. Nitu, "Natural language understanding for logical games," *arXiv preprint arXiv:2110.00558*, 2021.

[8] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "AllenNLP: A deep semantic natural language processing platform," *arXiv:1803.07640*, 2018.

[9] D. Toniuc and A. Groza, "Climebot: An argumentative agent for climate change," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2017, pp. 63–70.

[10] Z. Bakhtiari, H. [van Ditmarsch], and U. Rivieccio, "Bilattice logic of epistemic actions and knowledge," *Annals of Pure and Applied Logic*, vol. 171, no. 6, p. 102790, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168007220300142

[11] A. Przegalinska, L. Ciechanowski, A. Stroz, P. Gloor, and G. Mazurek, "In bot we trust: A new methodology of chatbot performance measures," *Business Horizons*, vol. 62, no. 6, pp. 785 – 797, 2019.