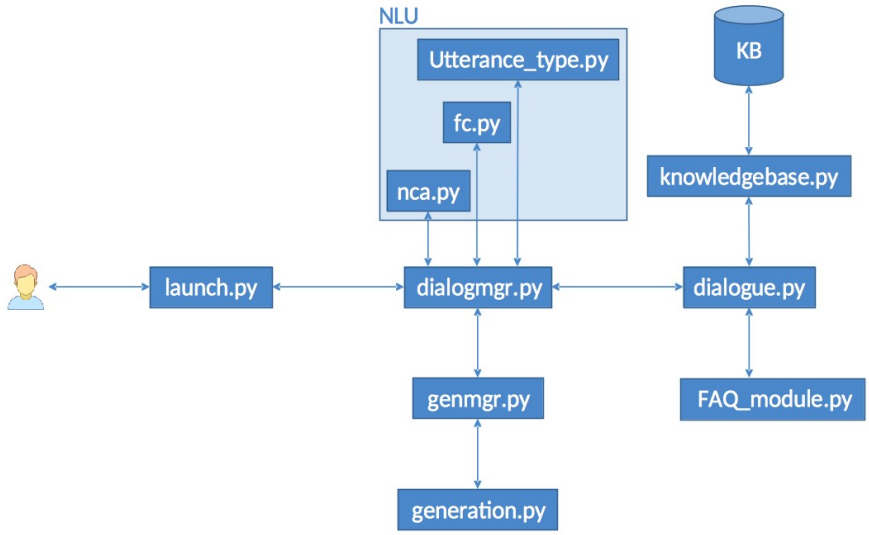
Exemple de déroulement du système de dialogue



Scenario 1: Main utterance type est Request

Utterance: « Give me a recipe of a soup »

```
[> give me a recipe of a soup                                                                                    ]
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O', 'O', 'O', 'B-F', 'O', 'O', 'B-C']
HC: slots: {'recipe_type': ['soup']}
HC: utterance's focus: ['recipe']
HC: utterance's context: ['soup']
preprocess utt : give me a recipe of a soup
HC: utterance's types: {'inform': ['give me a recipe of a soup']}
HC: utterance's main type: request
HC: answer's type: list of recipes
context: None
EC: slots: {'recipe_type': ['soup']}
EC: routing: request
```

Choix de la recette: « Alexandra soup »

```
# 161 recipes correspond to your search, including : "Agnelotti in Brodo" , "Alexandra soup" , "Apple Parsni
p Soup" . Which one do you want?
[> Alexandra soup                                                                                                ]
```

Chercher les ingrédients :

```
# OK, for Alexandra soup . Do you want the ingredients?
> yes
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['B-F']
True
HC: slots: {}
HC: utterance's focus: ['yes']
HC: utterance's context: []
preprocess utt : yes
HC: utterance's types: {'affirm': ['yes']}
HC: utterance's main type: affirm
HC: answer's type: None
context: context_recipe
EC: slots: {}
EC: routing: sub_dialogue_recipe_affirm

# The ingredients are : - 1 cup (240 ml) lentils
- 1 onion, diced
- 1 tablespoon (15 ml) butter
- 2 heaped tablespoons (30 ml) barley
- 2 tomatoes, peeled, seeded and diced
- 3 cups (720 ml) water
- salt and pepper, to taste
```

Chercher la procédure :

```
 Do you want the procedure?
> yes
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['B-F']
True
HC: slots: {}
HC: utterance's focus: ['yes']
HC: utterance's context: []
preprocess utt : yes
HC: utterance's types: {'affirm': ['yes']}
HC: utterance's main type: affirm
HC: answer's type: None
context: context_recipe
EC: slots: {}
EC: routing: sub_dialogue_recipe_affirm

# The procedure is : - Soak the lentils and barley together overnight (optional). Drain.
- Fry the onion in the butter until soft.
- Add the lentils, barley and water.
- Cook until the lentils are tender, (will take longer if grains are not soaked).
- Add the tomatoes when the soup is almost ready.
- Purée with a food mill, blender, or food processor.
- Reheat and adjust the seasoning.
 Anything else?
```

Scenario 2: Main utterance type est Quantitatif

Utterance: « How much oil in a pie ? »

```
> how much oil for a pie
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O', 'O', 'B-F', 'O', 'O', 'B-C']
HC: slots: {'ingredient': ['oil'], 'recipe_type': ['pie']}
HC: utterance's focus: ['oil']
HC: utterance's context: ['pie']
preprocess utt : how much oil for a pie
HC: utterance's types: {'quantitative': 'how much oil for a pie'}
HC: utterance's main type: quantitative
HC: answer's type: quantity
context: context_confirm_single_recipe
EC: slots: {'ingredient': ['oil'], 'recipe_type': ['pie']}
EC: routing: confirm_single_recipe_quantitative
routing: Error: no 'confirm_single_recipe_quantitative' dialogue function.
EC: slots: {'ingredient': ['oil'], 'recipe_type': ['pie']}
EC: routing: sub_dialogue_recipe_quantitative
routing: Error: no 'sub_dialogue_recipe_quantitative' dialogue function.
EC: slots: {'ingredient': ['oil'], 'recipe_type': ['pie']}
EC: routing: quantitative
context: None

# For this recipe you need 1 cup (240 mL) oil, preferably olive oil . Do you want to continue with Albanian
Vegetable Pie
>
```

Récupérer les ingrédients de cette recette

```
# OK, for Albanian Vegetable Pie . Do you want the ingredients?
> yes
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['B-F']
True
HC: slots: {}
HC: utterance's focus: ['yes']
HC: utterance's context: []
preprocess utt : yes
HC: utterance's types: {'affirm': ['yes']}
HC: utterance's main type: affirm
HC: answer's type: None
context: context_recipe
EC: slots: {}
EC: routing: sub_dialogue_recipe_affirm

# The ingredients are : - ½ cup (120 mL) chopped green onions
- 1½ packets (or about 30) pastry leaves (filo dough)
- 1½ pounds (680 g) spinach, chopped
- 1½ t (7.5 mL) salt
- 1 cup (240 mL) diced feta cheese
- 1 cup (240 mL) oil, preferably olive oil
- 2-3 eggs
```

Ne pas récupérer la procédure

```
 Do you want the procedure?
> no
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O']
HC: slots: {}
HC: utterance's focus: []
HC: utterance's context: []
preprocess utt : no
HC: utterance's types: {'negate': ['no']}
HC: utterance's main type: negate
HC: answer's type: None
context: context_recipe
EC: slots: {}
EC: routing: sub_dialogue_recipe_negate
acts:  {'act': 'ask_next'}
# Anything else?
>
```

Scenario 3: Main utterance type est Temporal

Utterance: « How much time does it take to prepare a cake ? »

```
[> how much time does it take to prepare a cake ?                                                ]
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O', 'O', 'O', 'O', 'O', 'B-F', 'O', 'O', 'O', 'B-C', 'O']
HC: slots: {'recipe_type': ['cake']}
HC: utterance's focus: ['take']
HC: utterance's context: ['cake']
preprocess utt : how much time does it take to prepare a cake
HC: utterance's types: {'temporal': 'how much time does it take to prepare a cake'}
HC: utterance's main type: temporal
HC: answer's type: duration
context: context_recipe
EC: slots: {'recipe_type': ['cake']}
EC: routing: sub_dialogue_recipe_temporal

# Time : prep: 20 minutes, baking: 60 minutes . Do you want to continue with 1-2-3-4 Cake
> 
```

Scenario 4: Main utterance type est Procedural

Utterance: « How to prepare a pizza ? »

```
[> how to prepare a pizza ?                                                                      ]
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O', 'O', 'O', 'O', 'B-C', 'O']
HC: slots: {'recipe_type': ['pizza']}
HC: utterance's focus: []
HC: utterance's context: ['pizza']
preprocess utt : how to prepare a pizza
HC: utterance's types: {'procedural': 'how to prepare a pizza'}
HC: utterance's main type: procedural
HC: answer's type: text_explanation
context: context_confirm_single_recipe
EC: slots: {'recipe_type': ['pizza']}
EC: routing: confirm_single_recipe_procedural
routing: Error: no 'confirm_single_recipe_procedural' dialogue function.
EC: slots: {'recipe_type': ['pizza']}
EC: routing: sub_dialogue_recipe_procedural
routing: Error: no 'sub_dialogue_recipe_procedural' dialogue function.
EC: slots: {'recipe_type': ['pizza']}
EC: routing: procedural
context: None

# The procedure is : - In a mixing bowl, mix warm water, salt and honey. Start off with extra warm water so
that it is still at the proper temperature for yeast.
- With the honey and salt fully dissolved, stir in the yeast and set aside for 5 minutes.
- Add 1 cup of flour and stir completely.
- Keep adding flour until dough begins to form a ball and is not sticky.
- Now knead dough thoroughly for a minute then cover bowl with lid.
- Let rise 1 hour. Then punch dough down. Let dough rise again for 1 1/2 - 5 hours.
- Preheat oven to 475°F (245°C) (add 20 minutes if using a pizza stone)
- Shape or freeze (See below)
- If using crust immediately, add desired toppings.
- Bake 15-25 minutes.
 Do you want to continue with Basic Pizza Crust
```

Scenario 5: Main utterance type est Description

Utterance: « what is the serving of pancake ? »

```
> What is the serving of pancake ?
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
<class 'list'> ['O', 'O', 'O', 'B-F', 'O', 'B-C', 'O']
HC: slots: {'recipe_type': ['pancake']}
HC: utterance's focus: ['serving']
HC: utterance's context: ['pancake']
preprocess utt : what is the serving of pancake
HC: utterance's types: {'description': 'what is the serving of pancake'}
HC: utterance's main type: description
HC: answer's type: text_description
context: context_confirm_single_recipe
EC: slots: {'recipe_type': ['pancake']}
EC: routing: confirm_single_recipe_description
routing: Error: no 'confirm_single_recipe_description' dialogue function.
EC: slots: {'recipe_type': ['pancake']}
EC: routing: sub_dialogue_recipe_description
context: context_recipe

# Serving : 8 4-inch pancakes . Do you want to continue with Almond Pancakes
>
```

Utterance: « what are the instructions for pizza ?» :

- De type description mais puisque le mot « instructions » dans le focus est synonyme de procédure c'est la fonction procedural qui est exécutée.
- Un dictionnaire de synonymes est utilisé pour traiter les cas de détection du type description mais faisant références à d'autres type, si le mot clé (ex: ingredient, component, constituent, … faisant référence à ingrédient) est dans focus.

```
> what are the instructions for pizza ?
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
/people/touzari/miniconda3/envs/lihlith/lib/python3.6/site-packages/torch/nn/functional.py:1350: UserWarning
: nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.
  warnings.warn("nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.")
<class 'list'> ['O', 'O', 'O', 'B-F', 'O', 'B-C', 'O']
HC: slots: {'recipe_type': ['pizza']}
HC: utterance's focus: ['instructions']
HC: utterance's context: ['pizza']
preprocess utt : what are the instructions for pizza
HC: utterance's types: {'description': 'what are the instructions for pizza'}
HC: utterance's main type: description
HC: answer's type: text_description
context: None
EC: slots: {'recipe_type': ['pizza']}
EC: routing: description
```

Scenario 6: Main utterance type est causal, propositional ou pas de réponse trouvée dans la kb le module FAQ est executé

Utterance: « what is the difference between Baking Powder and Baking Soda ? »

```
[> what is the difference between Baking Powder  and Baking Soda ?                                                    ]
The current process just got forked. Disabling parallelism to avoid deadlocks...
To disable this warning, please explicitly set TOKENIZERS_PARALLELISM=(true | false)
/people/touzari/miniconda3/envs/lihlith/lib/python3.6/site-packages/torch/nn/functional.py:1350: UserWarning
: nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.
  warnings.warn("nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.")
<class 'list'> ['O', 'O', 'O', 'B-F', 'O', 'B-C', 'I-C', 'O', 'B-C', 'I-C', 'O']
HC: slots: {'preparation_technique': ['baking', 'baking']}
HC: utterance's focus: ['difference']
HC: utterance's context: ['Baking Powder', 'Baking Soda']
preprocess utt : what is the difference between baking powderand baking soda
HC: utterance's types: {'description': 'what is the difference between baking powderand baking soda'}
HC: utterance's main type: description
HC: answer's type: text_description
context: None
EC: slots: {'preparation_technique': ['baking', 'baking']}
EC: routing: description
```

Les paragraphes retournés par le module IR

_ Recuperer les 5 premiers paragraphes de DoQA.
_ Recuperer les 5 premiers paragraphes de Cookbook Files.
_ Trier les paragraphes selon le nombre de mots en commun avec le contexte puis selon leurs scores.
_ Retourner les 3 meilleurs paragraphes quand le score est supérieur à un seuil prédéfini.

```
----------------------------
The top ranked paragraphs :
----------------------------
-->> ['Information about substitutions Baking Powder  and Baking Soda  Use a mixture of 2 parts Cream of Tar
tar  to 1 part baking soda as a substitute for baking powder.  It is sometimes possible to use baking powder
 in place of baking soda, however the presence of other acidic agents in the recipe can cause unexpected rea
ctions, and the quantity required (use four times as much baking powder as the recipe calls for baking soda)
 may leave a slightly unpleasant taste. ', 10.190435, 'Cookbook_files', 4]

-->> ['The Other Culinary Uses of Baking Soda As a Component in Baking Powder of baking soda is  Baking soda
 is often an ingredient in Baking Powder , along with an acidic ingredient like cream of tartar, and another
 gas-releasing chemical that activates only when subjected to heat. Baking powder loses its leavening power
very slowly, unless exposed to moisture.  ', 10.057763, 'Cookbook_files', 4]

-->> ['Information about baking powder Baking powder is a leavening agent composed of Baking Soda  and one o
r more acidic salts, such as tartaric acid (Cream of Tartar ) or dicalcium phosphate dihydrate.  As the baki
ng powder dissolves, the baking soda reacts with the acids to produce carbon dioxide gas bubbles, which are
trapped by the dough  around them.  Baking powder is used in recipes where the pH is 7, or neutral, and pure
 baking soda would fail to produce bubbles.', 9.972396, 'Cookbook_files', 4]
```

Les réponses retournées par le module ConvQA

```
# Here's what I found in the documents : - Information about substitutions Baking Powder and Baking Soda Use
 a mixture of 2 parts Cream of Tartar to 1 part baking soda as a substitute for baking powder
- Information about substitutions Baking Powder and Baking Soda
- Use a mixture of 2 parts Cream of Tartar to 1 part baking soda as a substitute for baking powder
- a mixture of 2 parts Cream of Tartar to 1 part baking soda as a substitute for baking powder
- Information about substitutions Baking Powder and Baking Soda Use
- Baking powder loses its leavening power very slowly, unless exposed to moisture
- The Other Culinary Uses of Baking Soda As a Component in Baking Powder of baking soda is Baking soda is of
ten an ingredient in Baking Powder
- The Other Culinary Uses
- Baking soda is often an ingredient in Baking Powder
- Baking soda is often an ingredient in Baking Powder , along with an acidic ingredient like cream of tartar
- Information about baking powder Baking powder is a leavening agent composed of Baking Soda and one or more
 acidic salts
- Information about baking powder Baking powder is a leavening agent composed of Baking Soda
- As the baking powder dissolves, the baking soda reacts with the acids to produce carbon dioxide gas bubble
s, which are trapped by the dough around them
- Baking powder is used in recipes where the pH is 7, or neutral, and pure baking soda would fail to produce
 bubbles
- a leavening agent composed of Baking Soda and one or more acidic salts Anything else?
>
```

Utterance: « How can I tell when my chicken is cooked ? »

Les paragraphes retournés par le module IR

```
---------------------------
The top ranked paragraphs :
---------------------------
-->> ["If your chicken burns, the issue is likely not that you cooked it too long, but that you cooked it to
o hot. How long to cook it depends on the total amount of meat, the size of pieces that it is in (anything f
rom bite sizes pieces for a stir fry up to an entire chicken), and how you're cooking it - oven, saute pan,
braising ...Roasting in the oven or braising in a stew are generally the most ignorable techniques - if you
overcook a roast it may dry out, but won't burn, and if you overcook a stew the meat (especially chicken) ma
y dissolve into shreds but burning is less likely.Since undercooked chicken can be unsafe, you should choose
 cooking techniques that are sure to fully cook the chicken without burning it. CANNOTANSWER", 5.5772595, 'D
oQA', 2]

-->> ["If your chicken burns, the issue is likely not that you cooked it too long, but that you cooked it to
o hot. How long to cook it depends on the total amount of meat, the size of pieces that it is in (anything f
rom bite sizes pieces for a stir fry up to an entire chicken), and how you're cooking it - oven, saute pan,
braising ...Roasting in the oven or braising in a stew are generally the most ignorable techniques - if you
overcook a roast it may dry out, but won't burn, and if you overcook a stew the meat (especially chicken) ma
y dissolve into shreds but burning is less likely.Since undercooked chicken can be unsafe, you should choose
 cooking techniques that are sure to fully cook the chicken without burning it. CANNOTANSWER", 5.5772595, 'D
oQA', 2]

-->> ['Information about cold water candy test When sugar syrup is cooked, water boils away resulting in a h
igh temperature sugar concentrate. Hence, as the temperature raises so does the concentration of sugar withi
n the solution. The nifty thing about cooking sugar is that the highest temperature it reaches can tell you
a little something about how it will turn out once it cools down to room temperature.Often it is practical t
o use a Candy Thermometer  how ever using the cold water test can be more fun (and dangerous). Note that the
 following temperatures are listed at sea level, as they are listed in most cookbooks, because of this actua
lly using the cold water test can be a more reliable gage when making candy. Each test is completed by dropp
ing a dollop of the hot candy solution in cold water (not room temperature), and then extracting the candy f
ormed. Be sure to allow for a moment for the candy to cool before handling it to avoid getting burnt. ', 6.7
305045, 'Cookbook_files', 1]
```

Les réponses retournées par le module ConvQA

```
# Here's what I found in the documents : - How long to cook it depends on the total amount of meat, the size
 of pieces that it is in
- anything from bite sizes pieces for a stir fry up to an entire chicken), and how you're cooking it
- If your chicken burns, the issue is likely not that you cooked it too long, but that you cooked it too hot
- How long to cook it depends on the total amount of meat, the size of pieces that it is in
- anything from bite sizes pieces for a stir fry up to an entire chicken), and how you're cooking it
- If your chicken burns, the issue is likely not that you cooked it too long, but that you cooked it too hot
- Information about cold water candy test When sugar syrup is cooked, water boils away resulting in a high t
emperature sugar concentrate
- Each test is completed by dropping a dollop of the hot candy solution in cold water (not room temperature)
, and then extracting the candy formed Do you want to know something else?
>
```

Score

queryWeight = boost (requête) * idf * queryNorm

boost (requête) = augmentation du champ au moment de la requête
Implication: les coups dans les domaines avec un boost plus élevé obtiennent un score plus élevé
Justification: un terme du champ A pourrait être plus pertinent que le même terme du champ B

idf = fréquence inverse du document = mesure de la fréquence d'apparition du terme dans l'index de ce champ
implémentation: log (numDocs / (docFreq + 1)) + 1
Implication: plus l'occurrence d'un terme dans différents documents est élevée, plus son score est bas
Justification: les termes courants sont moins importants que les plus rares

numDocs = le nombre total de documents dans l'index, sans compter ceux qui sont marqués comme supprimés mais qui n'ont pas encore été purgés. Il s'agit d'une constante (la même valeur pour tous les documents de l'index).

docFreq = le nombre de documents de l'index contenant le terme dans ce champ. Il s'agit d'une constante (la même valeur pour tous les documents de l'index contenant ce champ)

queryNorm = facteur de normalisation pour que les requêtes puissent être comparées
mise en œuvre: 1 / sqrt (sumOfSquaredWeights)
Implication: n'impacte pas la pertinence de ce résultat
Justification: queryNorm n'est pas lié à la pertinence du document, mais tente plutôt de rendre comparables les scores entre différentes requêtes. Cette valeur est égale pour tous les résultats de la requête

fieldWeight = le score d'un terme correspondant au champ
implémentation: tf * idf * fieldNorm

tf = fréquence du terme dans un champ = mesure de la fréquence à laquelle un terme apparaît dans le champ
implémentation: sqrt (freq)
Implication: plus un terme est fréquent dans un domaine, plus son score est élevé
Justification: les champs qui contiennent plus d'un terme sont généralement plus pertinents

freq = termFreq = nombre de fois où le terme apparaît dans le champ de ce document

fieldNorm = impact d'un hit dans ce champ
implémentation: lengthNorm * boost (index)

lengthNorm = mesure de l'importance d'un terme en fonction du nombre total de termes dans le champ
mise en œuvre: 1 / sqrt (numTerms)
Implication: un terme correspondant à des champs avec moins de termes a un score plus élevé
Justification: un terme dans un domaine avec moins de termes est plus important qu'un terme avec plus

numTerms = nombre de termes dans un champ

boost (index) = boost du champ au moment de l'index
Implication: les coups dans les domaines avec un boost plus élevé obtiennent un score plus élevé
Justification: un terme du champ A pourrait être plus pertinent que le même terme du champ B

maxDocs = le nombre de documents dans l'index, y compris ceux qui sont marqués comme supprimés mais qui n'ont pas encore été purgés. C'est une constante (la même valeur pour tous les documents de l'index)
Implication: (probablement) ne joue pas de rôle dans le calcul du score

coord = nombre de termes de la requête trouvés dans le document (omis si égal à 1)
mise en œuvre: recouvrement / maxOverlap
Implication: parmi les termes de la requête, un document qui contient plus de termes aura un score plus élevé
Justification: les documents qui correspondent aux termes les plus facultatifs obtiennent le score le plus élevé
chevauchement = le nombre de termes de requête correspondant dans le document
maxOverlap = le nombre total de termes dans la requête

FunctionQuery = peut être n'importe quel type de fonction de classement personnalisé, auquel le résultat est ajouté ou multiplié par le score de classement par défaut.
Implication: divers