Revisiting NLTK-DRT

DRT extension for NLTK updated and improved

Fedor Sizov[1] Ivan Rygaev[2]

[1]Saarland University

[2]Eberhard Karl University of Tübingen

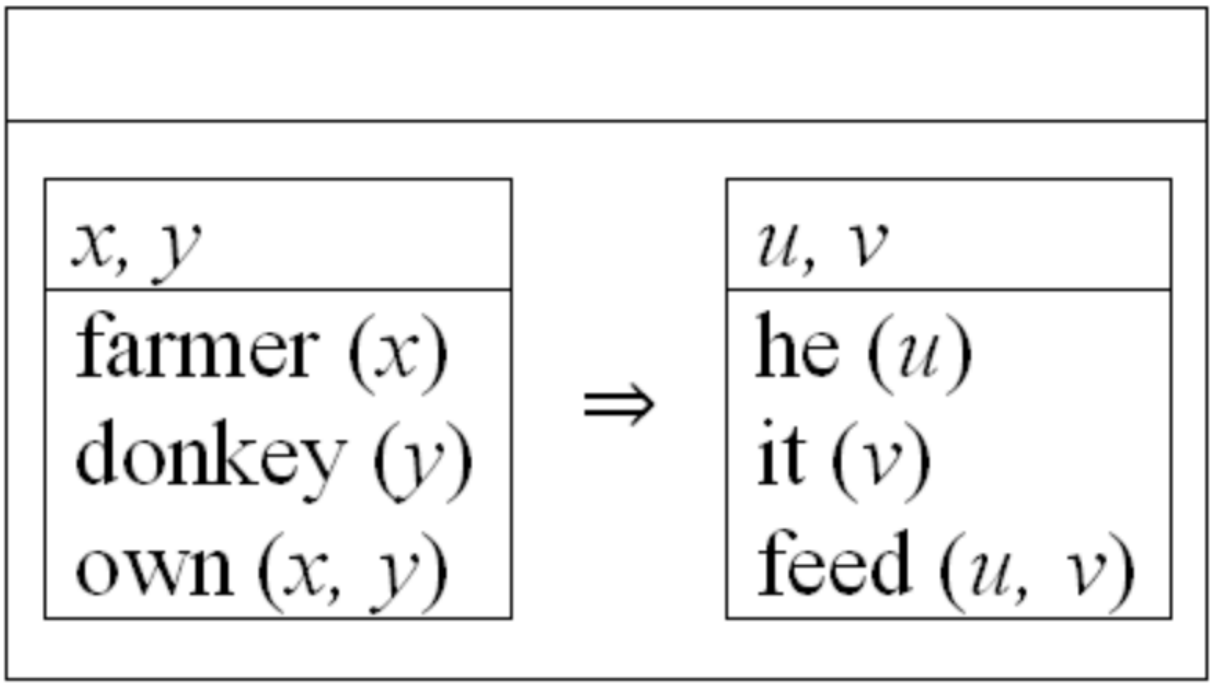
33th TaCoS conference, 16-18 May 2024

Table of Contents

Table of Contents

Discourse Representation Theory

- DRT is a dynamic semantic theory (Kamp 1981)
  - Employs a semantic representation called DRS (Discourse Representation Structure)
  - DRS consists of discourse referents and conditions
  - For complex sentences, a DRS can contain sub-DRSs
- *If a farmer owns a donkey, he feeds it*



Discourse Representation Theory: anaphora

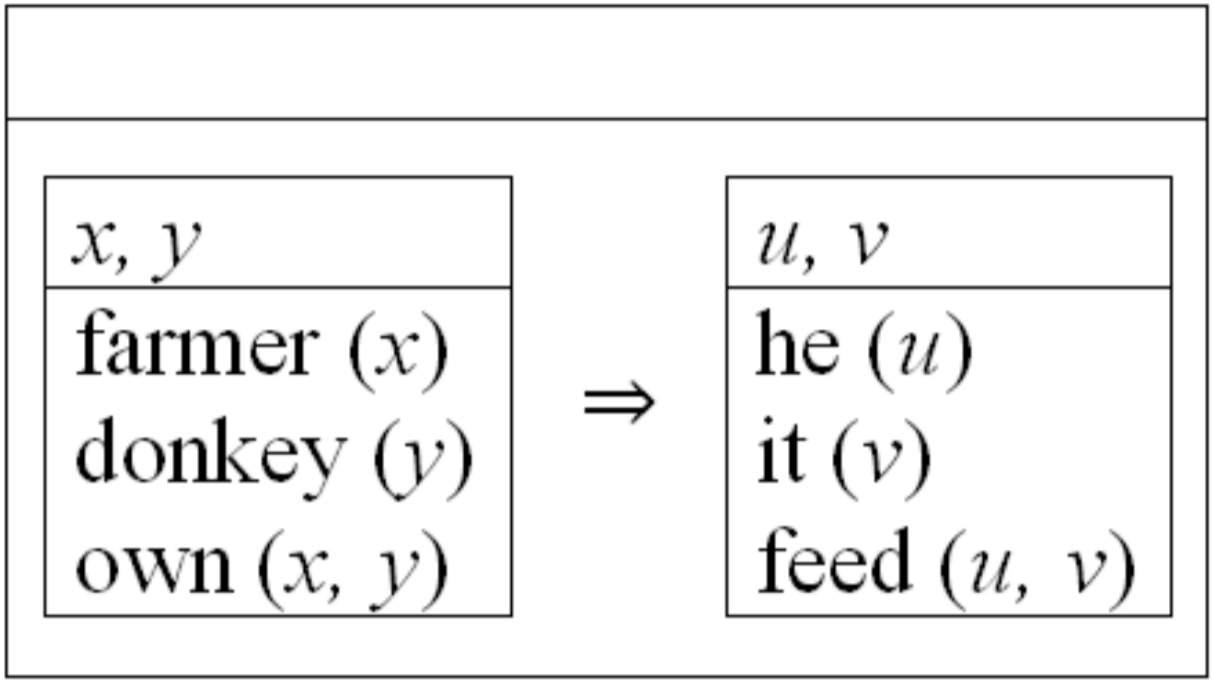
- DRT can be used for **anaphora** resolution

Discourse Representation Theory: anaphora

- DRT can be used for **anaphora** resolution
- Consider again the sentence *If a farmer owns a donkey, he feeds it*

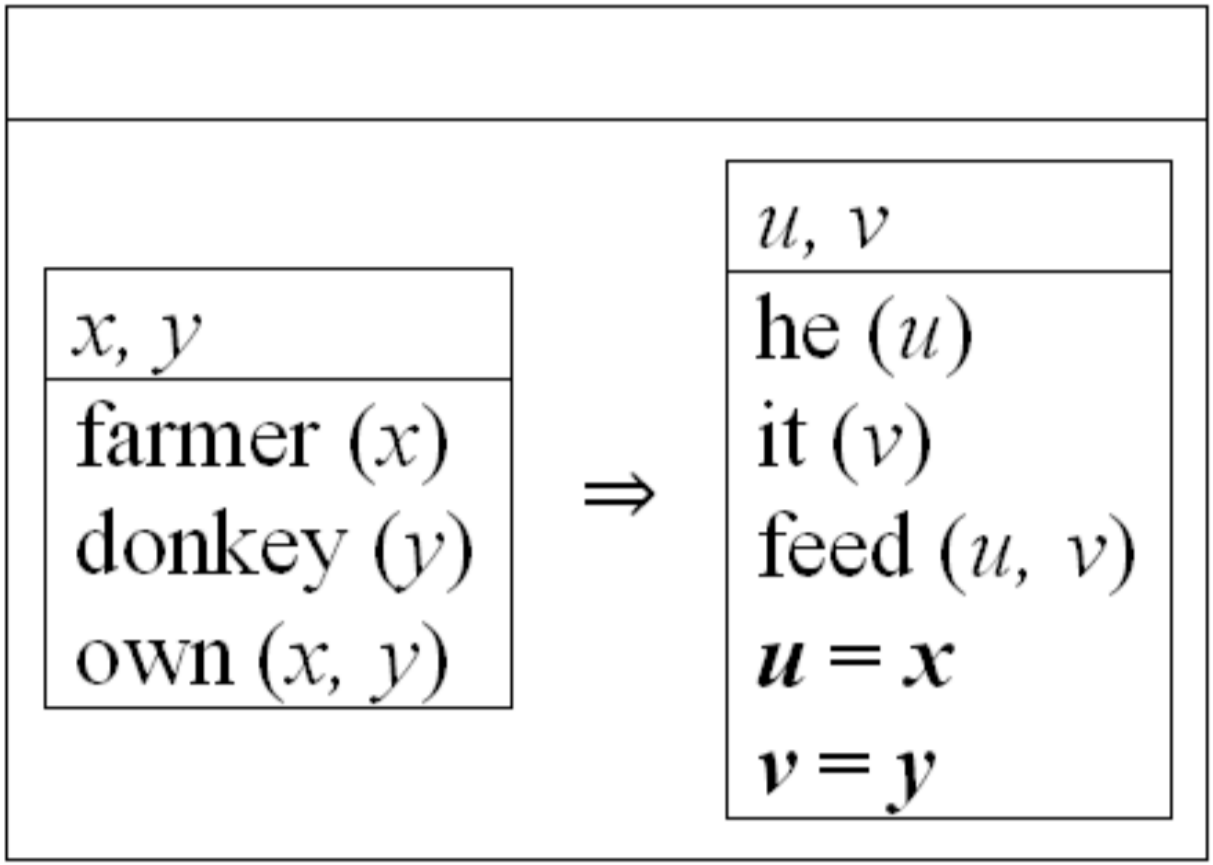
Discourse Representation Theory: anaphora

- DRT can be used for **anaphora** resolution
- Consider again the sentence *If a farmer owns a donkey, he feeds it*
- Step **1** (sentence with a non-resolved anaphora):



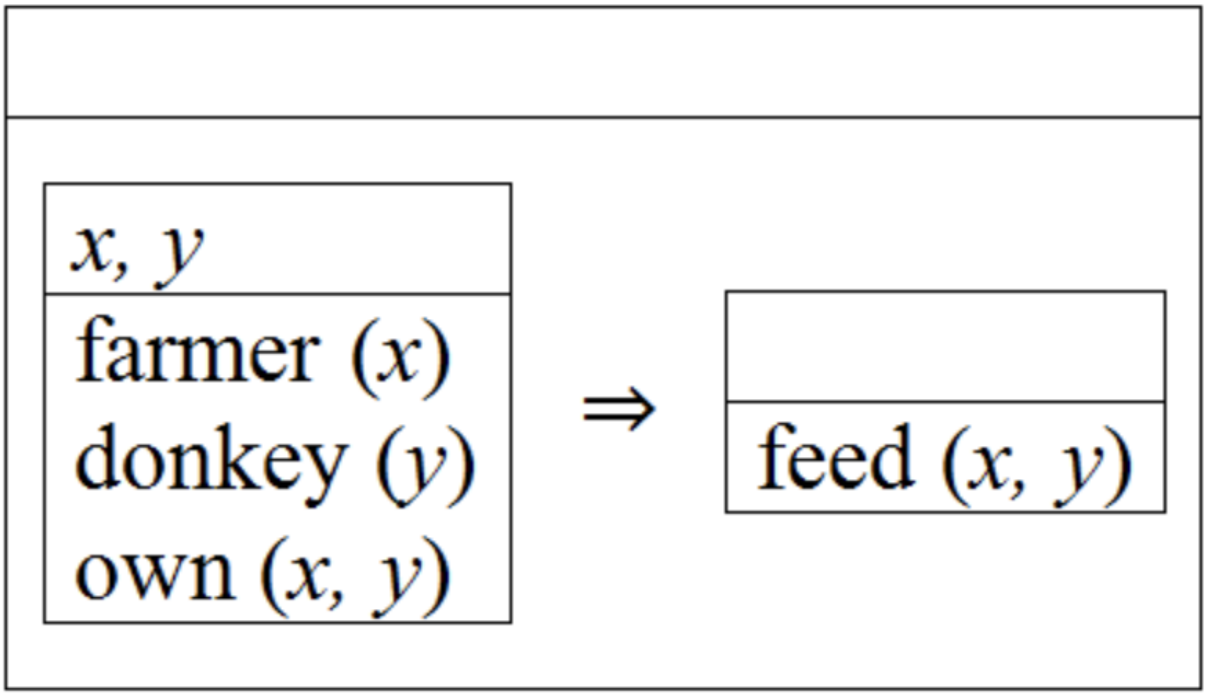
Discourse Representation Theory: anaphora

- DRT can be used for **anaphora** resolution
- Consider again the sentence *If a farmer owns a donkey, he feeds it*
- Step **2** (finding antecedents):



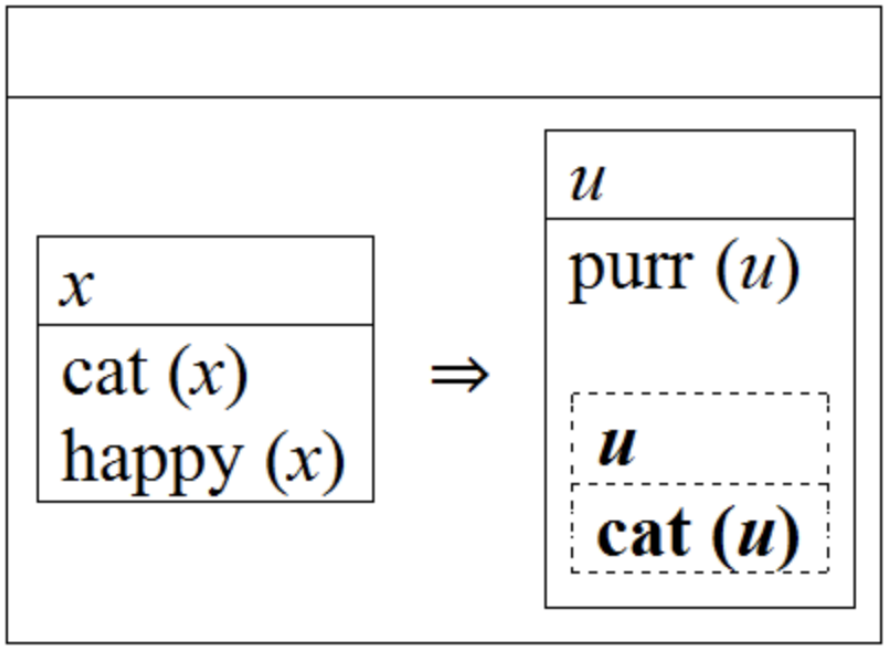
Discourse Representation Theory: anaphora

- DRT can be used for **anaphora** resolution
- Consider again the sentence *If a farmer owns a donkey, he feeds it*
- Step **3** (resolved):



Discourse Representation Theory: presupposition

- Binding theory of presupposition was introduced in (van der Sandt 1992)
  - A special sub-DRS (A-DRS) stores the presupposition content
  - A Preliminary DRS is a DRS with non-empty A-DRSs
- **Presupposition projection can be viewed as anaphora resolution**
- *If a cat is happy*, **the cat** *purrs* (A-DRS is non-empty)



Discourse Representation Theory: presupposition

- Preliminary DRS vs. Proper DRS
  - A-DRSs must be resolved
  - Once they are resolved, Main DRS becomes a Proper DRS
- *If a cat is happy, the cat purrs* (A-DRS is resolved)

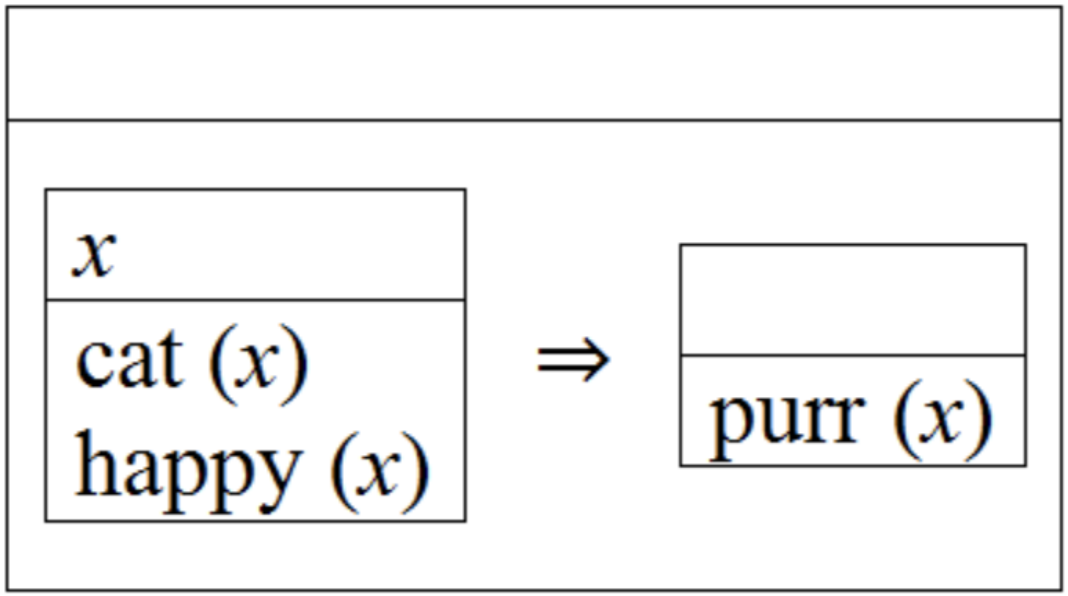


Table of Contents

DRT machinery in NLTK

- The **Natural Language Toolkit** (**NLTK**)[1] is a suite of libraries and programs for symbolic and statistical NLP in Python.

[1] `https://www.nltk.org/`

DRT machinery in NLTK

- The **Natural Language Toolkit** (**NLTK**)[1] is a suite of libraries and programs for symbolic and statistical NLP in Python.
- It supports, among other things, semantic reasoning functionalities.

[1] `https://www.nltk.org/`

DRT machinery in NLTK

- The **Natural Language Toolkit** (**NLTK**)[1] is a suite of libraries and programs for symbolic and statistical NLP in Python.
- It supports, among other things, semantic reasoning functionalities.
- NLTK has been supporting DRT since a long time ago; it contains a built-in basic anaphora resolution machinery

[1] `https://www.nltk.org/`

DRT machinery in NLTK

- The **Natural Language Toolkit** (**NLTK**)[1] is a suite of libraries and programs for symbolic and statistical NLP in Python.
- It supports, among other things, semantic reasoning functionalities.
- NLTK has been supporting DRT since a long time ago; it contains a built-in basic anaphora resolution machinery
- However, it **does not support** presupposition projection as anaphora resolution as in (van der Sandt 1992)

[1] `https://www.nltk.org/`

Table of Contents

Previous work on extending DRT in NLTK

- (Kislev, Makarov, Li 2010) developed a third-party module **NLTK-DRT**, which added support of presupposition projection as anaphora resolution (van der Sandt 1992) and time semantics to the basic DRT module of NLTK

Previous work on extending DRT in NLTK

- (Kislev, Makarov, Li 2010) developed a third-party module **NLTK-DRT**, which added support of presupposition projection as anaphora resolution (van der Sandt 1992) and time semantics to the basic DRT module of NLTK
- This third-party extension was not officially integrated into nltk codebase at the time of development

Previous work on extending DRT in NLTK

- (Kislev, Makarov, Li 2010) developed a third-party module **NLTK-DRT**, which added support of presupposition projection as anaphora resolution (van der Sandt 1992) and time semantics to the basic DRT module of NLTK
- This third-party extension was not officially integrated into nltk codebase at the time of development
- Neither was it later maintained and updated

Previous work on extending DRT in NLTK

- (Kislev, Makarov, Li 2010) developed a third-party module **NLTK-DRT**, which added support of presupposition projection as anaphora resolution (van der Sandt 1992) and time semantics to the basic DRT module of NLTK
- This third-party extension was not officially integrated into nltk codebase at the time of development
- Neither was it later maintained and updated
- As of today it was **no longer usable** due to multiple incompatibilities with nltk

Table of Contents

Addressing issues in NLTK-DRT: compatibility with NLTK

- NLTK-DRT extension had not been updated/maintanted since 2010

Addressing issues in NLTK-DRT: compatibility with NLTK

- NLTK-DRT extension had not been updated/maintanted since 2010
- Originally it was written in Python 2, which NLTK was also using at the time

Addressing issues in NLTK-DRT: compatibility with NLTK

- NLTK-DRT extension had not been updated/maintanted since 2010
- Originally it was written in Python 2, which NLTK was also using at the time
- We have done work on migration of NLTK-DRT codebase to Python 3

Addressing issues in NLTK-DRT: compatibility with NLTK

- NLTK-DRT extension had not been updated/maintanted since 2010
- Originally it was written in Python 2, which NLTK was also using at the time
- We have done work on migration of NLTK-DRT codebase to Python 3
- Also, since 2010, NLTK library has undergone many changes in structure of its submodules, which are imported by NLTK-DRT, and we had to take this into account when doing refactoring

Addressing issues in NLTK-DRT: implementation mistakes

- Some components of presupposition resolution machinery contained mistakes
- This led to the fact that most of the tests (written by the authors themselves) failed for the codebase

Addressing issues in NLTK-DRT: implementation mistakes

- Some components of presupposition resolution machinery contained mistakes
- This led to the fact that most of the tests (written by the authors themselves) failed for the codebase
- We carried out code analysis and fixed these mistakes, thus making 100% of the tests work

Addressing issues in NLTK-DRT: implementation mistakes

- Some components of presupposition resolution machinery contained mistakes
- This led to the fact that most of the tests (written by the authors themselves) failed for the codebase
- We carried out code analysis and fixed these mistakes, thus making 100% of the tests work
- Also, we migrated test suite itself (which contained 81 test) to the pytest framework

We define presupposition types based on FCFG

- There are different types of presupposition supported by NLTK-DRT:
  - definite descriptions
  - proper names
  - pronouns

We define presupposition types based on FCFG

- There are different types of presupposition supported by NLTK-DRT:
  - definite descriptions
  - proper names
  - pronouns
- Each type of presupposition should handled somewhat differently

We define presupposition types based on FCFG

- There are different types of presupposition supported by NLTK-DRT:
  - definite descriptions
  - proper names
  - pronouns
- Each type of presupposition should handled somewhat differently
- However, there are other types of presupposition which are not supported yet, for example, factive verbs

We define presupposition types based on FCFG

- There are different types of presupposition supported by NLTK-DRT:
  - definite descriptions
  - proper names
  - pronouns
- Each type of presupposition should handled somewhat differently

- Initially, the processing strategy for each type of presupposition was **hardcoded** inside the codebase

We define presupposition types based on FCFG

- There are different types of presupposition supported by NLTK-DRT:
  - definite descriptions
  - proper names
  - pronouns
- Each type of presupposition should handled somewhat differently

- It would be difficult for contributors to add new types of presupposition without deep code analysis

We define presupposition types based on FCFG

- **Solution:** define presupposition types outside of the code using FCFG (Feature Context-Free Grammar)
- Therefore, we have added the ability to define processing schemes for various presupposition types using FCFG files

Table of Contents

Future work

- As mentioned before, NLTK-DRT still does not support some types of presupposition, for example:
  - it-clefts (**It** *was her mother that always came up with excuses.*)
  - factive verbs (*I* **realized** *that I need to improve my skills.*)

Future work

- As mentioned before, NLTK-DRT still does not support some types of presupposition, for example:
  - it-clefts (**It** *was her mother that always came up with excuses.*)
  - factive verbs (*I* **realized** *that I need to improve my skills.*)
- In future, we would like to add support for these

Future work

- We have published our work on GitHub [2]
- At the moment, although we have made NLTK-DRT compatible with NLTK, this extension is still not officially included in NLTK module

[2] `https://github.com/sfedia/nltk-drt`

Future work

- We have published our work on GitHub [2]
- At the moment, although we have made NLTK-DRT compatible with NLTK, this extension is still not officially included in NLTK module
- NLTK already has basic support for DRT and anaphora resolution

[2] `https://github.com/sfedia/nltk-drt`

Future work

- We have published our work on GitHub [2]
- At the moment, although we have made NLTK-DRT compatible with NLTK, this extension is still not officially included in NLTK module
- NLTK already has basic support for DRT and anaphora resolution
- Therefore, an important task will be to add presuppositions support functionality directly to NLTK, merging NLTK-DRT extension and NLTK codebase

[2] `https://github.com/sfedia/nltk-drt`

Future work

- We have published our work on GitHub [2]
- At the moment, although we have made NLTK-DRT compatible with NLTK, this extension is still not officially included in NLTK module
- NLTK already has basic support for DRT and anaphora resolution
- Therefore, an important task will be to add presuppositions support functionality directly to NLTK, merging NLTK-DRT extension and NLTK codebase
- We aim to prepare a high quality pull request to NLTK (it will take some work to embed the extension inside nltk.sem.drt) and we will be glad if anyone wants to join us in this

[2] `https://github.com/sfedia/nltk-drt`

Future work

- We have published our work on GitHub [2]
- At the moment, although we have made NLTK-DRT compatible with NLTK, this extension is still not officially included in NLTK module
- NLTK already has basic support for DRT and anaphora resolution
- Therefore, an important task will be to add presuppositions support functionality directly to NLTK, merging NLTK-DRT extension and NLTK codebase
- We aim to prepare a high quality pull request to NLTK (it will take some work to embed the extension inside nltk.sem.drt) and we will be glad if anyone wants to join us in this
- Thank you for your attention!

[2] `https://github.com/sfedia/nltk-drt`