Google Summer of Code 2019

Apertium

Project Proposal: Anaphora Resolution and Long Distance Dependencies

Personal Details

Name: Tanmai Khanna

E-mail address: khanna.tanmai@gmail.com , tanmai.khanna@research.iiit.ac.in

IRC: khannatanmai

GitHub: khannatanmai

LinkedIn: khannatanmai

Current Designation: Undergraduate Researcher in the LTRC Lab, IIIT Hyderabad (III year student) and a Teaching Assistant for Linguistics courses

Time Zone: GMT+5:30

About Me

Open Source Softwares I use: I have used Apertium in the past, Ubuntu, Firefox, VLC.

Professional Interests: I'm currently studying NLP and I have a particular interest in Linguistics and NLP tools, specifically Machine Translation and its components.

Hobbies: I love Parliamentary Debating, Singing, and Reading.

What I want to get out of GSoC

I've enjoyed using Apertium in various personal and academic projects and it's amazing to me that I get an opportunity to work with them.

NLP is my passion, and I would love to work with similarly passionate people at Apertium, to develop tools that people actually benefit from. This would be an invaluable experience that classes just can't match.

I am applying for GSoC, as the stipend would allow me to dedicate my full attention to the project during the 3 months.

Why is it that I am interested in Apertium and Machine Translation?

Apertium is an Open Source Rule-based MT system. I'm a researcher in the IIIT-H LTRC lab, currently working on Machine Translation and it interests me because it's a complex problem which tries to achieve something most people believe is only achievable by humans. Translating data to other languages, and especially low-resource languages gives the speakers of those languages access to valuable data and can help in several domains, such as education, news, judiciary, etc. Machine Translation is often called NLP-Complete by my professors, i.e. it uses most of the tools NLP has to offer and hence if one learns to create good tools for MT, they learn most of Natural Language Processing.

Each part of Apertium's mission statement, especially the fact that they focus on Low Resource Languages, excites me to be working with them. While recent trends lean towards Neural Networks and Deep Learning, they fall short when it comes to resource-poor languages. Anaphora Resolution without complex linguistic information is a challenge that I'll be tackling during this Summer of Code with Apertium.

A tool which is rule-based and open source really helps the community with language pairs that are resource-poor and gives them free translations for their needs and that is why I want to work on improving on it.

I want to work with Apertium and GSoC so I can contribute to an important Open Source Tool while also honing my own skills, and I hope to become a part of this amazing community of developers!

Project Proposal

Which of the published tasks am I interested in? What do I plan to do?

Anaphora Resolution - Apertium currently uses default male. I wish to build a perfect Anaphora Resolution system, but that is obviously not possible in 3 months or without a lot of linguistic information so I propose we limit the scope of this project to use indicative features to pick an antecedent as a way to increase the fluency and intelligibility of output significantly.

While the problem is known as Anaphora Resolution, the task is to resolving long distance dependencies needed for **correct translations**. This includes pronoun resolution when it depends on the gender and/or number of the antecedent. Generally Anaphora Resolution includes resolving all referents to entities and verbs but not all of those are significant while doing general Machine Translation.

The formalism for the Anaphora Resolution tool will be language agnostic and there will be rules which should work for most related languages. Pronouns are present in a lot of languages and they change based on their antecedent's gender, number, person, etc. The aim is to figure out the antecedent and choose the correct pronoun accordingly.

Sentences like *The group agreed to release his mission statement*, *The girl ate his apple* are grammatically incoherent (if the discourse is limited to this) and an incorrect pronoun will more often than not confuse people in complex sentences.

What puts people off Machine Translation is the lack of fluency, and this tool will definitely generate more fluent sentences leading to more trust in this tool.

NOTE: Anaphora Resolution is one part of resolving long-distance dependencies. The method of resolving this will not be limited to anaphora and can be used for general coreference, agreement and other long-distance dependencies which need to identify the antecedent.

This project has promising future prospects:

- Adding Language Specific metrics for better identification
- Extending system to general coreference needed for MT
- Better Gisting translation

I want to be a contributor to Apertium even after GSoC and would love to continue this project forward after this summer.

Proposed Improvements

I will be testing this tool with Spanish-English and Catalan-English pairs while developing the tool.

However, the features will be made largely language agnostic and I will also try to evaluate how well it works for other language pairs which may need Anaphora Resolution, such as:

- French
- Russian
- Turkish
- Galician

Ultimately, the system should be able to do Anaphora Resolution for the following:

Possessive Pronouns

Spanish Sentence: *La chica comió su manzana*

Apertium Translation: *The girl ate his apple*

After Anaphora [Proposed Translation]: *The girl ate **her** apple*

Zero Pronouns

Eg. 1. Spanish Sentence: *Canta bien*

Apertium Translation: *It sings well*

After Anaphora [Proposed Translation]: ***He/She/It** sings well* (Based on context)

Eg. 2. Spanish Sentence: *La chica está aquí, lleva un vestido rojo*

Apertium Translation: *The girl is here, spends a red dress*

After Anaphora [Proposed Translation]: *The girl is here,* ***she*** *spends/wears a red dress*

Reflexive Pronouns

Spanish Sentence: *Se mató*

Apertium Translation: *It killed*

After Anaphora [Proposed Translation]: *He/She killed himself/herself*

Long Distance Agreement

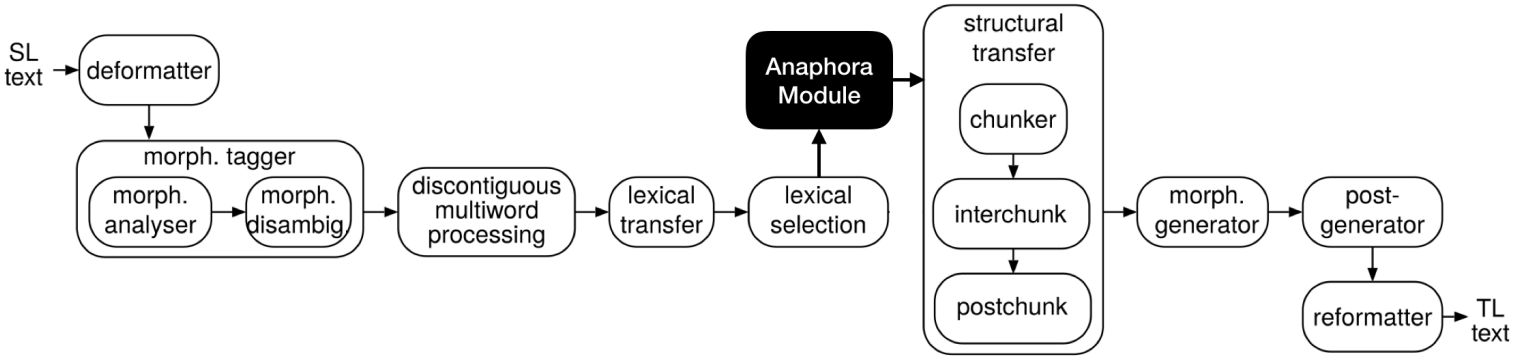
English Sentence: *The table is here. It is red.*

Apertium Translation: *La mesa es aquí. Es rojo.*

Proposed Translation: *La mesa es aquí. Es* ***roja.***

Any other long distance dependencies that can be detected

Idea Description



Modified Apertium Pipeline for Anaphora Resolution

As shown in the pipeline above, the Anaphora Module will take the Lexical Selection output as input and will use the proposed method (see below) to identify the antecedent of a word if it's anaphoric.
Then, it will attach the Lexical Unit of the antecedent to the LU where it is referred to (pronouns/adjectives/demonstratives).
After this, it will pass through transfer and we can add transfer rules to pick the correct pronoun/adjective based on the antecedent, if needed.

Working Example

Input Sentence

```
Els grups del Parlament han mostrat aquest dimarts el seu suport al batle d'Alaró…
```

Lexical Selection Output (Input to Anaphora Module)

```
^El<det><def><m><pl>/The<det><def><m><pl>$ ^grup<n><m><pl>/group<n><pl>$ ^de<pr>/of<pr>$
^el<det><def><m><sg>/the<det><def><m><sg>$ ^Parlament<n><m><sg>/Parliament<n><sg>$
^haver<vbhaver><pri><p3><pl>/have<vbhaver><pri><p3><pl>$
^mostrar<vblex><pp><m><sg>/show<vblex><pp><m><sg>$
^aquest<det><dem><m><sg>/this<det><dem><m><sg>$ ^dimarts<n><m><sp>/Tuesday<n><ND>$
^el seu<det><pos><m><sg>/his<det><pos><m><sg>$ ^suport<n><m><sg>/support<n><sg>$
^a<pr>/at<pr>$ ^el<det><def><m><sg>/the<det><def><m><sg>$
^*batle/*batle$ ^de<pr>/of<pr>$ ^*Alaró/*Alaró$^.<sent>/.<sent>$
```

Proposed Anaphora Module Output

```
^El<det><def><m><pl>/The<det><def><m><pl>/$ ^grup<n><m><pl>/group<n><pl>/$ ^de<pr>/of<pr>/$
^el<det><def><m><sg>/the<det><def><m><sg>/$ ^Parlament<n><m><sg>/Parliament<n><sg>/$
^haver<vbhaver><pri><p3><pl>/have<vbhaver><pri><p3><pl>/$
^mostrar<vblex><pp><m><sg>/show<vblex><pp><m><sg>/$
^aquest<det><dem><m><sg>/this<det><dem><m><sg>/$ ^dimarts<n><m><sp>/Tuesday<n><ND>/$
^el seu<det><pos><m><sg>/his<det><pos><m><sg>/group<n><pl>$
^suport<n><m><sg>/support<n><sg>/$
^a<pr>/at<pr>/$ ^el<det><def><m><sg>/the<det><def><m><sg>/$
^*batle/*batle/$ ^de<pr>/of<pr>/$ ^*Alaró/*Alaró/$^.<sent>/.<sent>/$
```

Part of Anaphora Module Output

```
^el seu<det><pos><m><sg>/his<det><pos><m><sg>/group<n><pl>$
```

Applying Anaphora Transfer Rules

```
^el seu<det><pos><m><sg>/their<det><pos><m><pl>/group<n><pl>$
```

Final Output without Anaphora Module

```
The groups of the Parliament have shown this Tuesday his support at the *batle of *Alaró…
```

Final Output with Anaphora Module

```
The groups of the Parliament have shown this Tuesday their support at the *batle of *Alaró…
```

Reference: Anaphora Resolution(Apertium)

Identifying Antecedents (Anaphora Module)

Since Apertium deals with low resource language pairs, this module will use very basic linguistic information: POS tags, gender, number information as opposed to parse trees, which require a lot more data to be accurate.

The module will assign **salience scores** to all antecedents in the context window.

This tool will not deal with First/Second Person pronouns. Here's why:

- ➡ For most languages only Third Person pronouns are differentiated by gender. Since our task is MT-based Anaphora Resolution, we can leave alone First and Second person as we can translate them correctly without resolving them.
- ➡ First and Second Person pronouns are largely resolved in the real world and including them in this tool will produce worse results than leaving them alone.

Possessive Pronouns

For possessive pronouns, we will use saliency scores from the factors mentioned below (Lappin and Leass Algorithm, Mitkov's Algorithm, Baldwin's Salience Features) and the highest scored antecedent will be selected for the pronoun.

If the same pronoun is repeated several times, the antecedent of the first will be attached to all the repetitions (dealing with examples such as "John lost his family, his house, his job and his dignity").

Reflexive Pronouns

Chomsky's Binding theory (1981) says reflexive pronouns must have local antecedents, which make them easier for us to detect. For Eg.,

John(i) washed himself(i) [CORRECT]

John(i) asked Mary to wash himself(i) [INCORRECT]

These will be resolved by identifying the noun closest to the verb. This identification based on simple linguistic information can be done if we can ignore embedded clauses effectively.

Zero Pronouns

Zero Pronouns will be resolved by **literally assuming a pronoun before the verb and then resolving it** using the same way we resolve possessive pronouns.

Due to word order differences in languages, this might give different results for different types of languages. Hence, we can evaluate this method and restrict it to certain languages if needed.

NOTE: Currently verbs in Spanish without a pronoun are translated to "it + verb".

This design choice makes sense due to the existence of **Pleonastic Pronouns**, i.e. pronouns which don't refer to any entity. For eg., "It is raining".

However, once we start trying to resolve Zero Anaphora, we will have to also distinguish Pleonastic Pronouns from Non-Pleonastic Pronouns which is largely done by semantics.

Long Distance Agreement for Adjectives

Long distance agreement for adjectives will become easier when the nearest pronoun has been resolved by the previous steps.

Here we can also try to ignore embedded clauses and the problem becomes similar to Reflexive pronouns where we need to identify the subject of the verb, in this case, the verb that connects the noun to the adjective.

Existing Algorithms

There are several existing algorithms for Anaphora Resolution:

- Pronoun Resolution, Hobbs
- Anaphora Resolution by Antecedent Identification Followed by Anaphoricity Determination, RYU IIDA
- Rule-Based Pronominal Anaphora Treatment for Machine Translation, Sharid Loáiciga
- A Rule-based Pronoun Resolution System for French, François Trouilleux

However, all of these algorithms use Parse Trees or Machine Learning, both of which require a large amount of data to give accurate outputs.

The following algorithms use, or can be modified to use very simple linguistic information already available in the Apertium Pipeline to perform reasonable Anaphora Resolution:

Ruslan Mitkov's Antecedent Indicators

Boosting Indicators (Will be updated as I study the data and find more indicators)

- First NPs
- Lexical Reiteration: Lexically reiterated items are likely candidates for antecedents.
- Section Heading Preference
- Collocation Pattern Preference: This preference is given to candidates which have an identical collocation pattern with a pronoun.
- Immediate reference (if it is pronoun then its reference)
- Sequential Instructions

Impeding Indicators (Will be updated as I study the data and find more indicators)

- Indefiniteness
- Prepositional NPs: NPs which are part of a PP are penalized.

Reference : Multilingual Anaphora Resolution, Ruslan Mitkov

We can add more factors, such as detecting heads of NPs, ignoring NPs that are part of Relative clauses to make the system more accurate.

Baldwin's Salience Features to find Coreference

- Unique in Discourse

If there is a single possible antecedent i in the read-in portion of the entire discourse, then pick i as the antecedent: 8 correct, and 0 incorrect.

- Reflexive

Pick nearest possible antecedent in the read-in portion of the current sentence if the anaphor is a reflexive pronoun: 16 correct, and 1 incorrect.

- Possessive Pronoun

If the anaphor is a possessive pronoun and there is a single exact string match i of the possessive in the prior sentence, then pick i as the antecedent: 4 correct, and 1 incorrect.

- Unique Subject/ Subject Pronoun

If the subject of the prior sentence contains a single possible antecedent i, and the anaphor is the subject of the current sentence, then pick i as the antecedent: 11 correct, and 0 incorrect.

Reference : High Precision Coreference with Limited Knowledge and Linguistic Reference, Breck Baldwin

Lappin and Leass

This algorithm considers these following factors in resolving the antecedent of a pronoun:

- **Recency:** Entities introduced in recent utterances are more likely to be referred to by a pronoun than entities introduced in utterances further back.
- **Grammatical Role:** Entities introduced in subject position tend to get topicalised, and are more likely to be referred to by a pronoun than entities in object positions.
- **Repetition:** Entities that have already been referred to frequently are more likely to be pronominalised than those that have not.

  George needed a new car. His previous car got totalled, and he had recently come into some money. Jerry went with him to the car dealers. He bought a Nexus.

- **Parallelism:** Pronouns are more likely to refer to those entities that do not violate syntactically parallel constructions.

  John took Bill to the zoo; Mary took him to the park.

Algorithm

- Resolving each Pronoun from left to right
- Collect potential referents from up to 4 sentences back
- Filter out antecedents that don’t satisfy agreement/syntax constraints
- Select remaining antecedents with the highest salience value; add pronoun to class.

Saliency Factors

Salience Factor	Example	Weight
Current sentence		100
Subject emphasis	**John** opened the door	80
Existential emphasis	There was **a dog** standing outside	70
Accusative emphasis	John liked **the dog**	50
Indirect object	John gave a biscuit to **the dog**	40
Adverbial emphasis	*Inside **the house**, the cat looked on	50
Head Noun emphasis	**The cat** in the house looked on	80

- The salience of a referent is the sum of all applicable weights
- The salience of a referent is halved each time a sentence boundary is crossed
- This, along with the weight for being in the current sentence, makes more recent referents more salient
- Lappin and Leass report 86% accuracy for their algorithm on a corpus of Computer manuals

Even though **the original algorithm requires parse trees to detect these saliency features**, I'll be attempting to detect as many of them as I can for the chosen languages with the available linguistic features.

References:

Original Paper: An Algorithm for Pronominal Resolution

Anaphora Resolution, University of Cambridge

Work Plan

Community Bonding Period (May 6 - May 27)

- Understand the Apertium pipeline fully
- Modify and Understand individual files
- Get familiar with the files that I need to modify
- Formalise the problem, limit the scope of anaphora resolution (To Anaphora needed for MT)
- Flowchart of the proposed system
- Write Pseudocode for identifying Salience Factors
- **Study the EuroParl corpus** and see which anaphors the method will be able to resolve on paper
- Explore Constraint Grammar and use it if it proves to be beneficial

Week 1 (May 27)

- Automatic Annotation of anaphora for evaluation (EuroParl Corpus)
- Implement a preliminary scoring system for antecedent indicators [work for Spanish-English and Catalan-English for now]
- Decide on a definite context window

Week 2 (June 3)

- Implement Basic Anaphora for Possessive Pronouns in C++
- Create a transfer-pattern-like file as a way to mark possible NPs as antecedents.
- Implement transfer rules for Possessive Pronouns
- A basic prototype ready
- TEST the prototype with the pipeline

Week 3 (June 10)

- Implement Basic Anaphora for Reflexive Pronouns (On Verbs) [For Spanish and Catalan]
- Implement Basic Anaphora for Zero Pronouns (On Verbs) [For Spanish and Catalan]
- Implement transfer rules for the above
- TEST the system extensively
- Document the outline

Week 4 (June 17)

- Implement the system to work out all possible antecedents
- Add ability to give antecedents a score
- TEST basic sentences with single antecedents, Test the pipeline
- Test and Evaluate for Possessive, Reflexive, Zero Pronouns in Spa-Eng pair
- Test and Evaluate for Possessive, Reflexive, Zero Pronouns in Cat-Eng pair

Deliverable #1: Anaphora Resolution for single antecedents, with transfer rules [The full pipeline]

Evaluation 1: June 24-28

Week 5 (June 24)

- Implement Antecedent Indicators:
- Implement Code to Identify Boosting Indicators
- Implement Code to Identify Lappin and Leass Indicators (as many as possible)

Week 6 (July 1)

- Code to Identify Impeding Indicators
- Code to Identify Adjective Agreement Antecedent
- Implement transfer rules for agreement in adjectives for Cat-Eng & Spa-Eng
- Code to remember antecedents for a certain window
- Give scores to the antecedent indicators

Week 7 (July 8)

- Implement detection of remaining salience features
- Implement remaining transfer rules for anaphora in pronouns Cat-Eng & Spa-Eng
- Code Salience Indicators & Implement tie breaking systems
- Modify the scoring system based on performance in the pairs
- TEST system with French and Russian.

Week 8 (July 15)

- Implement fallback for anaphora (in case of too many antecedents or not past certainty threshold)
- TEST Scoring System
- TEST and Evaluate current system and produce precision and recall
- TEST and Evaluate Agreement for Adjectives
- TEST system with Turkish and any other required language pairs.
- Document Antecedent Indicators, Scoring System, Fallback for Cat-Eng & Spa-Eng

Deliverable #2: Anaphora Resolution with saliency features detection, scores, and a fallback mechanism

Evaluation 2: July 22-26

Week 9 [OPTIONAL: If current system not producing good enough results]

- Implement Expectation-Maximization Algorithm using monolingual corpus
- Compare with current system
- Test EM Algorithm and the implemented system

Week 9 [NOT OPTIONAL] (July 22)

- Implement code to ignore embedded clauses
- Evaluate increase in detection
- Insert into Apertium pipeline
- Implement code to accept input in chunks and process it

Week 10 (July 29)

- EXTENSIVELY TEST final system
- Test with French, Russian, Turkish, Galician, etc.
- Evaluate and find out which features are language agnostic
- Decide on list of features for agnostic anaphora and for language specific anaphora

Week 11 (August 5)

- Any remaining coding and improving the system
- TEST on multiple pairs and give Evaluation Scores
- TEST for backwards compatibility and ensure it

Week 12 (August 12)

- Wrap up on the final module
- Complete the overall documentation with observations and future prospects

Final Evaluations: August 19-26

Project Completed!

NOTE: Week 11 and Week 12 have extra time to deal with unforeseen issues and ideas

---

A description of how and who it will benefit in society

It will definitely benefit users of several language pairs and hopefully will attract more people to the tool. I'm from India and for a lot of our languages, we don't have the data to create reliable Neural MT systems. Similarly, for all resource poor languages, Apertium provides an easy and reliable MT system for their needs. That's how Apertium benefits society already.

However, what discourages people from using Machine Translation is unintelligible outputs and too much post editing which makes it very time consuming and costly for them. While Apertium aims to make minimal errors, as of now it selects a default male pronoun and that leads to several unintelligible outputs. Fixing that as much as possible and making the system more fluent and intelligible overall will encourage people to use Machine Translation and will reduce costs of time and money.

Reasons why Google and Apertium should sponsor it

I feel this project has a wide scope as it can affect almost all language pairs and helps almost everyone using Apertium. A decent Anaphora Resolution will give the output an important boost in its fluency and intelligibility, not just for one language, but all of them.

It's a project which has promising future prospects as well - it paves the way for general coreference resolution for low-resource languages and several language specific features can be added to improve it.

With this project I aim to help the users of Apertium, I wish to become a regular contributor to Apertium and become equipped to do a lot more Open Source Development in the future for other organisations as well.

By funding this project, Google will help improve an important Open Source tool and promote Open Source Development. In a world of Proprietary softwares, this is an invaluable resource for society and supports innovation that everyone can benefit from.

Skills and Qualifications

I'm currently a third year student at IIIT Hyderabad where I'm studying Computational Linguistics. It is a dual degree where we study Computer Science, Linguistics, NLP and more. I'm working on Machine Translation in the LTRC lab in IIIT Hyderabad and I'm part of the MT group in our university. We meet weekly to discuss issues in our Neural, Statistical and Rule-Based Machine Translation systems.

I've been interested in linguistics from the very beginning and due to the rigorous programming courses, I'm also adept at several programming languages like Python, C++, XML, Bash Scripting, etc. I'm skilled in writing Algorithms. Data Structures, and Machine Learning Algorithms as well.

I enjoy reading research papers and I have analysed several research papers in preparation for this proposal, all of which have been cited. I also have a lot of experience studying data which I feel is essential in solving any problem.

Due to the focused nature of our course, I have worked in several projects, such as building Translation Memory, Detecting Homographic Puns, POS Taggers, Grammar and Spell Checkers, Named Entity Recognisers, Building Chatbots, etc. all of which required a working understanding of Natural Language Processing. Most of these projects were done offline in my research lab and aren't available on GitHub because of the privacy settings but can be provided if needed.

I am fluent in English, Hindi and have basic knowledge of Spanish.

The details of my skills and work experience can be found here: CV

Coding Challenge

I successfully completed the coding challenge and proposed an alternate method to process the input as a chunk which resulted in a speedup of more than 2x.

The repo can be found at: https://github.com/khannatanmai/apertium

Files in Repo:

- Code to do basic anaphora resolution (last seen noun), input taken as a stream (byte by byte)
- Code to do basic anaphora resolution (last seen noun), input taken as a stream (as a chunk)
- Speed-Up Report

Non-Summer-Of-Code Plans

I will have my college vacations during GSoC so will have no other commitments in that period and will be dedicated to GSoC full time, i.e. 40 hours a week.

I am planning to go on a short trip to London from 30 May to 3rd June so I will start working for Week 1 earlier and finish it before the trip. I will still be able to devote 20 hours in Week 1.

I have also kept the workload lighter in Week 1 for the same reason.