Dudău Andrei, 246

Anaphora Resolution

Python Implementation in Romanian Language

About the Project

The purpose of this project was to implement Mitkov's Anaphora Resolution in Romanian Language. This was achieved using RACAI's TTL for the text parsing.

Problems with TTL

Implementing Mitkov's algorithm with RACAI's TTL proved to be a slightly more difficult task. One of the problems was lack of part of sentence tagging. The absence of this kind of information rose some challenges.

Another problem was proper nouns. TTL did not provide gender for those. In order to fix this issue, I created a web scraper, searched wikipedia articles and created two lists: one for female first names and one for male first names. They are attached inside my project and they may prove useful for other people using TTL.

Mitkov's Anaphora Resolution

On short, Mitkov's Algorithm requires the text split into sentences and the words need to be POST tagged. For each pronoun (anaphora found):

1. Calculate npInd for each NP that agrees in number & gender with the pronoun
2. Lower the score if the NP is definite
3. Lower the score if the NP is accompanied by a preposition
4. Increase the score if NP is the first in the sentence
5. Increase the score if NP immediately follows an indicative verb (discuss, illustrate, etc.)
6. Increase the score if NP is repeated in the paragraph (appears one or more times)
7. Increase the score if NP represents a term in the genre of the text
8. Increase the score if NP has an identical collocation pattern to the pronoun
9. Increase the score if NP is Immediate reference (it is resolving 'it')
10. Increase the score of an NP if it is closer to the pronoun

In my project, I did not implement item #7 as I did not implement the means to find out the genre of a text.

Classes used in the project

Ttl: namespacing class for calls to the TTL. This is where the networking logic was implemented. The requests and their parameters are available in this file. Those may also prove useful for others trying to implement something with TTL. Please note that in order to have correct POS tagging, diacritics are required. In order to use TTL with diacritics, it can not process unicode. It is required to first parse your text in SGML, apply TLL on the SGML text and then convert the SGML back to UTF8 with a third API call.

1. Utf8_to_sgml
2. Xces
3. sgml_to_utf8

WordParser: namespacing class for parsing results from TTL

XcesWord: a basic, general, word. All words are an XcesWord.

- Id
- Original
- Lemma
- Ana
- Type
- chunk

Word: special kind of XcesWord, inherits from it, has gender & number. All nouns and pronouns are parsed to this type of object

- Gender
- Number
- Proper
- Definite
- Preposition
- After_verb - used for #9 (resolving it)
- Intex_in_sentece - needed for #8 (collocatio pattern_
- Pattern - needed for #8 (collocatio pattern_

WordType: Enum for possible word types:

- NOUN = "N"
- VERB = "V"
- ADJECTIVE = "A"
- PRONOUN = "P"
- DETERMINER = "D"
- ARTICLE = "T"

- ADVERB = "R"
- ADPOSITION = "S"
- CONJUNCTION = "C"
- NUMERAL = "M"
- INTERJECTION = "I"
- RESIDUAL = "X"
- ABBREVIATION = "Y"
- PARTICLE = "Q"
- END = "."
- PUNCTUATION = ","
- PUNCTUATION2 = ";"
- PUNCTUATION3 = ":"
- SHORT_PRONOUN = "-"

Phrase: a sentence phrase (NounPhrase, VerbPhrase, etc.)

- Words
- Type
- Order
- Gender (if NP)
- Number (if NP)

Sentence: an object that holds an array of words & an array of phrases

- Words
- phrases

Candidate: a possible anaphora

- Word
- Np_ind
- Definite_score
- Preposition_score
- First_sentence
- First_sentence_score
- Reiteration_score
- Resolving_it_score
- Distance_score
- Recent
- Previous_word
- Multiple - when there are multiple words forming a NP

Anaphora: a pronoun and its candidates; various methods to compute the npInd score for each candidate

Mitkov: Objects that ties all together. Finds the pronouns, creates the anaphoras objects from provided text and calls the methods.

Main: Calls Mitkov's Algorithm on the provided text. Here you can edit the text you want to have analyzed.

Particularities observed for Romanian Language

1. Pronoun - Third Person, Masculine, Plural could also refer one or multiple feminine nouns, if a masculine one was also part of an NP
2. The verb with lemma "avea" tends to refer direct objects. NP following this verb could have their npInd increased.