20170396 예성현

The program is divided into 2 parts: looking at the pronoun containing sentence, looking at the context sentence. For snippet-context, this context is the prior sentence, and for page-context, this context is sentences in the whole Wikipedia page. The main idea is finding named entity subject. For many pronouns, they refer to the subject of the same or prior sentence. However, there might be exceptional cases that need to be handled. For example, in the sentence “Mr.Jones calls her.”, ‘Mr.Jones’ is not ‘her’ even though it is a named entity subject. To handle these cases, the condition ‘there has to be a NP chunk between candidate and the pronoun to be true’ is added. Also, candidates are divided into close, far candidate according to the distance between pronoun and the candidate. Because close candidates have higher probability to be the answer, it is assumed for some cases that if the close one is true, the far one is false. Moreover, close candidate is given priority in terms of test order. The grammar to parse sentences is nearly identical to the one from HW4. For a CLAUSE chunk, if there is a NNP POS tag in NP chunk inside it, it is interpreted as a subject.

If there is a candidate in the pronoun containing sentence, that sentence is considered first. If the pronoun is the first word of the sentence, the candidate is false because that pronoun would refer to the entity in the prior sentence. This also correctly detects direct citation sentence such as ‘Julia said, “She is too young.” ’ because it checks whether the first alphabet of pronoun is uppercase or not. Therefore, for this case, ‘She’ is not ‘Julia’. After named entity subject checking, if the subject is the candidate, that candidate is true. If the subject is something not the candidate, the candidate is false because it would be likely that the pronoun refers to the subject which is not the candidate. If there is no named entity subject, the case is more complicated. If the pronoun comes early, the candidate is false because it is true only when the candidate is a subject like the sentence “On the morning of her disappearance, Alison stayed with Ian”. If the pronoun comes late, only the close one would be true. Example sentence for this case would be “Instead of William, the director picked Jack because he was competent” with ‘he’ referring to ‘Jack’.

If the candidate containing sentence is prior to pronoun containing sentence, the implementation is different depending on the context. For snippet context, it only looks at the candidate containing sentence and do the similar step as the previous paragraph. For page context, if the pronoun is nominative case (‘he’ or ‘she’) and there is a named entity subject, it goes through the same process as the snippet context. The reason nominative case pronoun is the dividing point is because it is more likely that it would refer to the prior sentence’s subject. Otherwise, it references the Wikipedia page. From the whole text from the page, it searches the same pronoun. For the sentence containing the same pronoun, it searches the subject. The candidate which has more corresponding subject would be true and the other candidate would be false. If there is a named entity subject but not both, then the target candidate is false because it is likely that the pronoun would refer to the other named subject. However, if there is no named entity subject, the target candidate is true. This is reasonable inference because if there is no NE subject, it is likely that the pronoun refers to the candidate which is not the subject.

The overall snippet-context performance is recall: 58.2 precision: 63.0 f1: 60.5, while the page-context performance is recall: 63.9 precision: 65.9 f1: 64.9. Page context has about 4% increase compared to snippet context. Because page-context looks at the whole page context when the prior sentence does not have much information, more correct answer would be driven. The false–negative cases are slightly higher than false-positive cases. This is mainly because the priority given to close candidate. For some cases, if the close candidate is true, false candidate is automatically false, without

going through further process. Also, the bias ratio is nearly 1 because gender information is not considered.

The performance of the model is limited to some point due to the performance of the grammar. Without using any external models, the sentences are tagged and chunked solely on the grammar. Therefore, the same limitations that was mentioned on the previous homework assignment happen again. Especially, because the text is mainly about explanation of an object or a person in Wikipedia, there are many apposition structure sentences (sentence such as 'Tim, the teacher of school, gave a speech') and relative clauses. These parts are not interpreted as subject because they do not have a general CLAUSE like structure (NP+VP structure). Building grammar with only regex parser has some limitations when dealing with complex sentence structures like these. By using external models such as allennlp and corenlp parsers for chunking, these problems could be partially solved, although the runtime would be much longer. There were other trivial problems such as grammatically incorrect sentences, tokenization problems which was due to the informal feature of Wikipedia.

There were issues on checking NP chunk between pronoun and candidate. For the sentence, "Mary collaborate with her.", it is correct to infer that 'her' is not 'Mary' because there is no NP chunk in between. However, for sentence, "Mary is kind as her reputation proves it.", 'her' would refer to 'Mary' even though there is no NP chunk in between. This is because program only collects NP chunk, not VP chunk. If there is a VP chunk in between, it is most likely that the candidate and pronoun be subject-object relationship, returning the candidate false. Therefore, false negative could be reduced if VP chunk collect feature is added.

To improve the performance of snippet-context for the case when the candidate containing sentence is different from pronoun containing sentence, we can use the context of pronoun containing sentence as well. Especially, first order logic can be used to analyze the context. For example, for a discourse "James married to Ann and had a daughter named Alice. However, James divorced with her after meeting Emma.", 'her' is actually referring to 'Ann'. If we construct logic (James, Ann, married), (James, Alice, daughter), (James, divorced, her), 'her' would easily refer 'Ann' if meaning of divorced is considered (breaking up after marriage).

The performance could be further improved by considering gender. We know by experience that there are girl-like names and boy-like names. By training with validation set, we can get sets of typical girl names such as Emma, Annie and boy names like James, John. Also, the gender information could be inferred from the page context as well. Although, gender information cannot be an absolute standard, it would improve the performance to some degree.