Algorithm.docx -

**This word documents contains 3 algorithms for (i) pre-processing (ii) relation extraction (iii) post-processing stages of our framework (GIX).**

|  |  |
| --- | --- |
| Algorithm 1: Pre-processing | |
| Input: Ks = keywords; P = Database; Retmax = Maximum number of articles to be retrieved  Output: Sse2 = Selected set of sentences; Ese2 = Recognized set of named entities | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | **Function** retrieve\_articles  **Pass In:** Ks,  P, Retmax  Perform PubMed Search Query using Ks  **Pass out:** Retrieved Retmax abstracts  **End function**  **Function** Sentence\_tokenization  **Pass In:** Retmax abstracts  Tokenizers ← Load nltk punkt sentence tokenizer  **For** *i=1* to Retmax **Do**  Tokenize *ith* abstract in sentences using Tokenizers  **End For**  **Pass out:** Tokenized sentences (St); t ← total number of sentences  **End function**  **Function** Sentence\_eliminator\_1  **Pass In:** Tokenized sentences (St)  Modelse1 ← Load fine-tuned sentence eliminator 1 model  Tokenizerse1 ← Load BertTokenizer (biobert\_v1.1\_pubmed)  Sse1 ← Initialize an empty list to store selected sentences  **For** *i=1* to t **Do**  Tokenize *ith*sentence using Tokenizerse1  Convert tokenized sequence to ids  Generate attention masks for ids  Predict classification using Modelse1  **If** predicted class is not 0 **then**  Append Sse1 with *ith* Sentence  **End If**  **End For**  **Pass out:** Selected set of sentences Sse1; se1 ← total number of sentences  **End function**  **Function** Sentence\_eliminator\_2  **Pass In:** Selected set of sentences Sse1  Modelse2 ← Load fine-tuned BERN2 model  Sse2 ← Initialize an empty list to store selected sentences  Ese2 ← Initialize an empty list to store recognized named entities for each sentence  **For** *i=1* to se1 **Do**  Predict all gene/protein entities (*Ei*) present in *ith* Sentence using Modelse2  **If** length of *Ei* is greater than 2 **then**  Append Sse2 with *ith* Sentence  Append E se2 with predicted list of entities (*Ei*) in *ith* Sentence  **End If**  **End For**  **Pass out:** Selected set of sentences Sse2; Recognized named entities Ese2;  se2 ← total number of sentences  **End function** |
|  |  |

|  |  |
| --- | --- |
| Algorithm 2: Relation Extraction | |
| Input: Sse2 = Selected set of sentences; Ese2 = Recognized set of named entities  Output: R = List of extracted relations; = List of predicted confidence for each relation; | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | ***Sre***← Initialize an empty list to store labelled sentence for relation extraction  ***Rre*** ← Initialize an empty list to store entity pair for each sentence in *Sre*  **For** *i=1* to se2 **Do**  ***P* (*A*, *T*)** ← create entity pairs from list of recognised entities (*Ei*) in *ith*sentence  **For** *p=1* to *P* **Do**  *Sip* ← Replace first entity in the pair (*Ap*)with “$GENE\_AGENT#”  And second entity in the pair (*Tp*)with “$GENE\_TARGET#” in *ith* sentence  **For** x=1 to *Ei* **Do**  **If** *Ex* not in (*Ap*, *Tp*) **Do**  *Sip* ← Replace *Ex*with “BLANK” in sentence *Sip*  **End If**  **End For**  Append *Sre* with labelled sentence *Sip*  Append *Rre* with entity pair in labelled sentence (*Ap*, *Tp*)  **End For**  **End For**  Modelre ← Load fine-tuned relation extraction model  Tokenizerre ← Load BertTokenizer (biobert\_v1.1\_pubmed)  **For** *i=1* to *re* **Do**  Tokenize *ith*sentence in *Sre* using Tokenizerre  Convert tokenized sequence to ids  Generate attention masks for ids  Predict relation using Modelre  **If** predicted relation is not *None* **Do**  Append **R**with predicted entity pair from *Rre*  Append with predicted model confidence value;  ∈ (0, 1)  **End If**  **End For** |

|  |  |
| --- | --- |
| Algorithm 3: Post-processing | |
| Input: = List of extracted relations; = List of predicted confidence for each relation; γ = Threshold value  Output: = List of refined extracted relation;  = List of computed confidence factor for each relation | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | **Lun-w** ← Load the list of unwanted words  **ModelNER** ← Load fine-tuned BERN2 model  **For** *i=1* to R **Do**  **For** each entity in R **Do**  Ref(e) ← Declare an empty string to store the refined entity  Remove unwanted words (**Lun-w)** from the entity (e)  **For** each word in e **Do**  Recognize if the word is a gene/protein entity using **ModelNER**  **If** word is not a gene/protein **Do**  **If** word is not a typical English language word **Do**  Append Ref(e) with the word  **End If**  **Else Do**  Append Re(e) with the word  **End Else**  **End If**  Replace original entity e with refined entity (Ref(e)) in R  **End For**  **End For**  **For** *i=1* to R **Do**  **If** both refined entities in Ri are not **NULL** **Do**  Compute CF*i*using equation (i) for entity pair Ri  **If** CF*i*> γ **Do**  Append Re with entity pair Ri  Append CF with computed confidence CF*i*  **End If**  **End If**  **End For** |