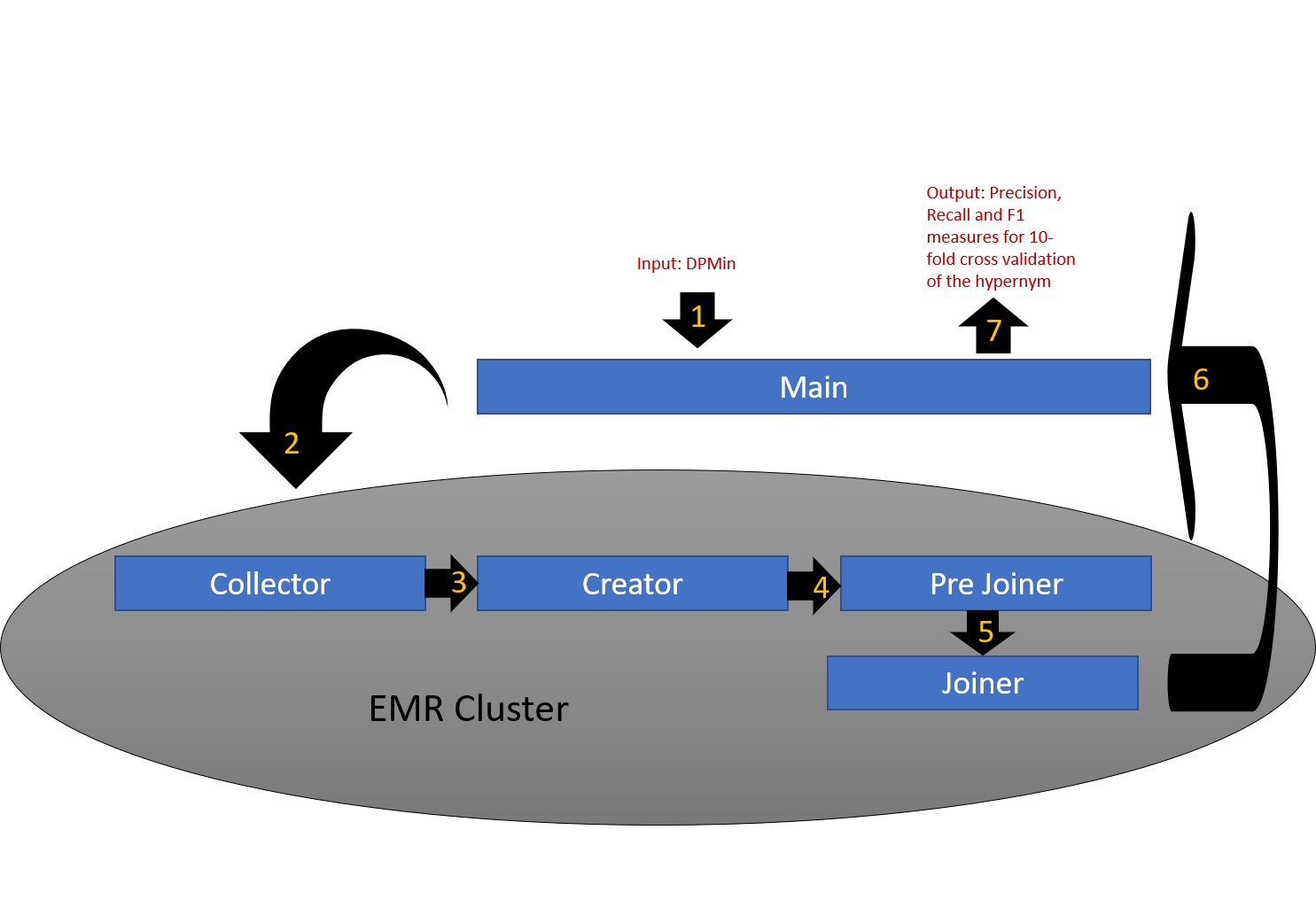
Introduction:

Following “Learning syntactic patterns for automatic hypernym discovery” article written by Rion Snow, given syntactic ngrams-biarcs00 and annotated set which represents whether there is a hypernymy relation between two words, we’ve been tasked to build a system which gets as an input <DPMin> and returns Precision, Recall and F1 measures for 10-fold cross validation of the hypernym annotated set. Our system is composed of 5 modules: 4 modules which stands as map-reduce steps and 1 module(main) that in charge of creating the EMR flow, running it, waiting for its completion and returning the needed output.

System Structure - Illustration:



System Structure - Modules:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Module | Task: | | | | | | | | |
| Main | **Input:** DPMin  **Output:** Precision, Recall and F1 measures for 10-fold cross validation of hypernym annotated set. | | | | | | | | |
| **Objective:**  Initializing Hadoop map-reduce work flow and waiting for the EMR cluster to complete, once it completed downloads the feature vectors from the cloud and generates all the relevant data using weka open source library. | | | | | | | | |
| Collector | **Input:** Syntactic ngrams – biarcs00  **Output:** <Key: nouns, Value: <dependency path, occurrences>>\* | | | | | | | | |
| **Objective:** | | | | Map: For each row at the corpus, computes the minimal dependency path between any two nouns, adding occurrences number and sending it to the reducer. | | | Reduce:  Checks whether a dependency path is valid in accordance with DPMin. | |
| Creator | **Input:** Annotated Set & Collector output  **Output:** <Key: dependency path, Value: <nouns, occurrences, annotation>>\* | | | | | | | | |
| **Objective:** | | | Map: Reading all inputs and sending them to the reducer such that “AnnotatedSet” value(true\false) will be the first value of pair of nouns if exist to arrive the reducer. | | | | | Reduce:  Sends only the nouns that exist on “Annotated Set”. |
| Pre Joiner | **Input:** Creator output  **Output:** <Key: nouns, Value: <dependency path , occurrences, annotation >>\* | | | | | | | | |
| **Objective:** | | Map: Sends the Keys & Values as is to the reducer. | | | | Reduce:  Counting the number of dependency paths between two nouns that are annotated (following the previous step) and index them accordingly. At cleanup sends the counter itself so it can be used at the next phase. | | |
| Joiner | **Input:** Pre Joiner output  **Output:** arff format for weka | | | | | | | | |
| **Objective:** | Map: Sends the Keys & Values as is to the reducer. | | | | Reduce:  Creates the arff file for weka using the indexed from previous step and adds all the relevant rows needed for the arff file to be valid | | | |