Extracting relations with NLTK

* To look at manpage open2, use command: man 2 open

Install NLTK

1. If pip command not found: sudo easy\_install pip
   1. If easy\_install not found: sudo apt-get install python-setuptools
2. Python
   1. Import nltk
   2. Nltk.download(‘punkt’)
   3. Nltk.download(‘averaged\_perceptron\_tagger’)

Links

* http://www.nltk.org/install.html

NLTK Steps and Tips

Working Version (relation\_extraction.py):

Entities

1. Extract IEntities(FirstEntities) based on .BI and manpage name.
2. Extract TPEntities(second entities) based on .TP: .B or .BR; and .B
3. Tokenize and tag the entities.
4. Use tag\_extraction function to make entities into iterable list.
5. Chunk label the entities.

Document

1. Use numbers of spaces to determine the tabs. Every time it indents, push onto stack. If it un-indents, pop from stack. Concatenate lines with the same indent. (Relation dictionary).
2. For each element in OuterStack, push that element’s entities and relations into an inner stack and pop them in the order of TPEntities, Relation, and IEntities.

Trial 1

The first step is to convert the unstructured data of the uncompressed man pages into structured data. We can do this by tokenizing the sentences and tagging each tokenized word using the nltk.pos\_tag() function. From there we then chunk each of the part of speech tags, which will further divide our data into smaller subsets. This will help us identify the named entities. However, this may not be necessary as I have already extracted the entities that we want into lists. The next step is to then extract the relations between the entities. We can use regular expressions to help us single out certain relations that we are looking for in particular. Lastly, we can create two for loops, one to parse the entire man page, and the second using the nltk.semextract\_rels( entity1, entity2, manpage, corpus, pattern). Hopefully this will help us extract the relations that we want.

1. Extract the two entities using own algorithm.
2. Use: <https://www.nltk.org/book/ch07.html#ref-ie-segment>
3. Start with chapter 7 figure 1.1
4. Then go to chapter 7 figure 3.2 (consecutiveNPChunkTagger):
   1. “train\_sents” is
   2. “history”: is a list of the tags that we’ve predicted for the sentence so far. Each tag in history corresponds with a word in sentence.
5. Change entity lists into word format??
   1. We can just use double bracket and treat whole entity list as one sentence (better)
   2. Or run the entity list through nltk word tokenizer.
6. Tag entity lists
7. Convert tagged entity lists to IOB format.
   1. Use regex expressions to do so.

Trial 2

Use nltk.sem.extract\_rels() code as reference.

Entities

1. Extract IEntities(FirstEntities) based on .BI and manpage name.
2. Extract TPEntities(second entities) based on .TP: .B or .BR; and .B
3. Tokenize and tag the entities.
4. Use tag\_extraction function to make entities into iterable list.
5. Chunk label the entities.

Man decompressed Documents

1. Use regex to fill in blank lines with “.RE”. This is to set a marker for end of paragraphs? (Might be a limitation for the algorithm).
2. Tokenize and tag the words in the document.
3. IEntities are labeled as “B-NP”.
4. TPEntities are labeled as “I-NP”.
5. For now relationships are determined based on word preceding a colon, or a variation of the word “follow”. They are labeled as “R”.
6. Use stack to extract/associate the right entities with the right relations.
   1. Only pop when .RE is found
   2. Only add entities and relations to dictionary when there is a relation.

\*Note: NLTK has format:

* Sentence: [‘S1’, ‘S2’]
* Word: [[‘W1 of S1’, ‘W2 of S1’], [‘W1 of S2’, ‘W2 of S2’]]
  + Basically a list within a list
* Tags: [[(‘W1 of S1’, ‘t’), (‘W2 of S1’, ‘t’)], [(‘W1 of S2’, ‘t’), (‘W2 of S2’, ‘t’)]]
* Chunks: [((‘W1 of C1’, ‘t’), (‘W2 of C1’, ‘t’), u’C’)]

Regex

* [] brackets means this or that. Ex) [89] means match 8 or 9.
* .\* means match anything. (. Means match anything put newline and \* means 0 or more).
* () parenthesis just groups things.

Unused Files:

* IOB\_format.py – puts the entities in IOB format?? Currently incorrect because should be whole document? (Trial 1).
* Relation\_extractionBR.py – extract the relations with BR method (Trial 2).
* Relation\_extractionB.py – extract the relations with B method (Trial 2).
* Manflags.py – outputs two documents, IEntities.txt and TPEntities.txt.
* Nltk-trainer-master – from someone’s github claiming this will help you extract

Final Notes:

* Some decompressed man pages won’t have any flags, so can’t extract anything.
* An alternative way may be to use IEPY for relationship extract, but be wary, it has few documentations when I last worked with it!
* Also, there is some strange anomaly with decompressing the man pages each time…are they changing??? …Or is my code just broken…my openRelation.txt for the last “modes are” is missing “O\_RDONLY”….Nvm found the issue is with NLTK. (Search for NLTK comment in code).

Links

* <http://www.nltk.org/book/ch07.html#ex-ie4>
* <https://www.youtube.com/watch?v=6j6M2MtEqi8&list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL&index=4>
* https://groups.google.com/forum/#!topic/nltk-users/Rhl7KZ4lE-4
* https://docs.google.com/document/d/1fTYpUXwfO3z4\_mj7taieuiV9VllF3IQb\_rdEeztYmEw/edit