Communication and Cognition (Comcog) Named Entity Recognition Methodology

# Introduction

# Assumptions

The author assumes that the reader is familiar with statistical and rule based Natural Language Processing (NLP) in general, in particular, stand-off techniques and tools including those that underly the UIMA and GATE software platforms, where the original text is not altered, but annotations, or markups are created that refer back to positional or character-based offsets within the original document. The author assumes the reader is familiar with NLP platforms that are built using a pipeline composed of *annotators*, where a representation of the input document and associated markups is passed from annotator to annotator. Each annotator adds, edits or deletes the markups that refer back to the original text. Each annotator handles a specific or atomic task that down-pipeline annotators rely upon. (This is the way of GATE and UIMA). Describing algorithms that utilize this technology stack are best described by the sequence of annotators in general, only delving into specific annotator algorithm details where needed.

For these task, the author assumes the reader is familiar with clinical medical terminologies used for Natural Language Processing, and in particular the Unified Medical Language System (UMLS), and the International Classification of Functioning Disability and Health and (ICF).

# Prior Related Software

# Dictionary Lookup and Lexicon Sources

The dictionary lookup process involves loading all lexica into a hash table. Framework uses a lookup mechanism that starts with the span of a sentence, works from right to left (yes, right to left), that first uses all the tokens as a key to be looked up in the hash. When failing, the left-most token is dropped and that key is looked up in the hash. This continues until a key is found. Once found, the tokens consumed change right side of the window to be searched. This insures maximal span matches, favoring matches where the head of a term is found. The English language is structured where the important parts of terms or the head a phrase or term is the last or right part. This right to left window technique favors capturing at least the important parts of terms when there is ambiguity.

Framework comes with and runs with the following default lexica (unless otherwise tuned)

* UMLS SPECIALIST 2022 Lexicon
* Document Titles
* CCDA Section Names
* Page Header and Page Footer Evidence
* Common Clinical Slot Names
* Date and Time Terms
* Labs Terminology (from LOINC)
* Blood Panel Terminology (from LOINC)
* Units of Measure Terminology
* Person Evidence Terms
* ~~USPS Address Terms~~
* ~~Clinical Demographics Terms~~

For the ComCog task, additional lexica were curated, with two approaches used to gather the additional terminology.

Terms thought to be the most general Comcog classes from the MF Ontology were gathered. Each of these were used as seed terms to traverse through the 2022 UMLS to gather all descendent terms to be incorporated into the terminology used for lookup. Communication and Cognition activities is poorly covered in the UMLS, and this process was disappointing, only picking up ICF Activities and Participation subclasses, for the most part.

It was recognized that communication and cognition activities evidence are likely to be verbs and adverbs. An effort was made to classify the University of Colorado’s VerbNet classes into IPIR and Comcog categories manually. That effort was expanded to have each of the Comcog categories (about 162 classes) to further be subclassified into the 20 subclasses manually. This was done by 3 domain expert annotators and one software engineer (me). This was a daunting task involving classifying over 5000 lemmas resulting from the 162 classes. There was reason for classifying at the lemma level rather than class level. The classes were too inclusive, with most classes involving lemmas (in VerbNet terminology: members) from the classes that didn’t really fit the semantic relatedness criteria we were classifying at. We had time constraints so the effort was a quick and rough effort with the understanding that we’d go back and do better curation when time allowed. For the most part, the VerbNet Comcog tagged lemmas is the terminology employed.

# Pipelines, Algorithms, and Rules

## Pre-processing Annotators

The IPIR NER built upon the Mental Functioning NER’s pipeline, which, in turn, is built upon a general NLP pipeline to segment text into constituent document-decomposition parts, which is built upon a UIMA infrastructure.

### Document Decomposition Pipeline

The Document Decomposition Pipeline identifies Sentences, Terms, Tokens, Section Names, Section Zones within the text.

#### Important: Term Lookup

Of note here: the secret sauce if you will: the terms identified are based first on dictionary/lexica lookup, where care has been taken to craft the content of the dictionaries, and where each of the dictionary entries contain term attributes including semantic categorization(s) for each term and the pedigree of where the term was sourced from. It is the semantic categorizations for each term identified within the text that is paramountly used down pipeline. While there are many ways to identify terms within the text, how and what is contained within the dictionaries is the salient part to each task.

### Mental Functioning NER Pipeline and Annotator

The mental functioning NER pipeline is a series of annotators to identify Mental Functioning Ontology (MFO) concepts within text. As such, it includes annotators that create Behavior Evidence, Emotion Evidence, and (financial, social, institutional) Support Evidence from the terms that have these semantic categories prior to an annotator that creates Mental Functioning Evidence and Mental Functioning Mention Markups. The Mental Functioning Ontology Annotator identifies and creates Mental Functioning Evidence Mark ups from terms that have semantic categories that generalize up to Mental Functioning within the MFO Ontology.

Specifically, a Mental Functioning Mention encapsulates the outer spans of any MFO evidence found within the scope of a sentence[1] or slot:value[2], or check-box[3]. Note that MFO mentions will not be made from within the bounds of a section name.

This is over-generative. Filters are applied to weed out spurious mentions. Underutilized but built in is a filter to filter out any mentions found within document types (identified from the document-decomposition pipeline) that indicate this document should not be processed. Currently, the pipeline does not identify any forbidden document types. In prior tasks, documents identified a-priori from their source HER were tagged as administrative or clinical, and those documents tagged as administrative were filtered out. The plan is when page and document classification become available, this feature will become important, particularly to ignore mentions that come from templated text coming from SSA forms found within the corpora we’ve been processing.

### Ad-hoc Filters

There are additional ad-hoc filters applied as well.

Documents that include SSA forms, where the templated text contains what look like mental functioning mentions, but are not, because the mentions are within templated text that are most likely to be instructions on how to fill out the form. Such mentions that included *see [x]* within the scope of the sentence are filtered out.

Similarly, mentions that can be attributed to the author of the document, not the client or patient are filtered out. These mentions have evidence that includes the use of pronouns like *we* and *our* along with use of pronouns like *you*, *your*, *his*, *her*, *she*, and *him*. Such mentions are also tagged with *Provider* attribution as a feature at the term and sentence level.

### Customization for inFACT Processing

The Mental Functioning Ontology pipeline is being embedded within the inFACT workflow, where it is important that the same segmentation is used across all the inFACT processing. As such, the functionality being described here assumes the input is GATE/Interlingua formatted files that include sentence mark-ups. An additional preprocessing annotator was added to the MFO pipeline that swaps in the inFACT sentences for the framework sentences. Also, upstream inFACT processing will have created *IPIR\_yes*, *IPIR\_no*, *Comcog\_yes*, and *Comcog\_no* annotations within the bounds of each sentence via trained LLM models.

For the purposes of inFACT, it is these inFACT labeled \_yes mark-up is then further processed down pipeline. The MFO evidence and mentions that overlays these markups are available for the downstream processing. Outside of inFACT, equivalent MFO markups that generalize up to IPIR and ComCog mentions have been generated and can be used independent of inFACT processing.

## The Comcog Sub-categorization Annotator

The Comcog Sub-Categorization annotator further sets 20 attributes to each Comcog\_yes.

For each Comcog\_yes sentence span, all the MFO generated evidence that generalizes to Comcog Activities from the Ontology, along with behavior and Support evidence that cover that span are gathered and if there is any of these found, further processing is done.

There are 20 attributes set:

* ICF d110\_d129\_Purposeful Sensory Experiences
* ICF d130\_d159\_Basic Learning
* ICF d160\_Focusing Attention
* ICF d163\_Thinking
* ICF d166\_Reading
* ICF d170\_Writing
* ICF d172\_Calculating
* ICF d175\_Solving Problems
* ICF d177\_Making Decisions
* ICF d179\_Applying Knowledge Other
* ICF d210\_d220\_UndertakingTasks
* ICF d230\_Carrying Out Daily Routine
* ICF d240\_Handling Stress
* ICF d310\_d329\_Receiving Communication
* ICF d330\_d349\_Producing Communication
* ICF d350\_d369\_Conversation and Discussion
* Adaptation
* Applied Memory
* Pacing
* Persistence
* There is a rule that if there was no evidence for an *Comcog\_yes* mention found, an *ICF d179\_Applying Knowledge Other* is created to satisfy that criteria.

### Non Relevant Evidence

There is a check to throw out mentions that are not IPIR, ComCog, or Mental Functioning related due to some known context. For example, if the sentence has the word *invoice*, or *billing* in it, this is not an MFO mention. There are 1389 terms marked as *NotMFO* which includes *case number*, *department of social services*, *social security act* along with terms like *swelling* and *please*.

These non-relevant evidence terms were garnered and added from frequency distributions of terms appearing in false positive mentions and never appearing in true positive or false negative mentions.

The Comcog annotator creates evidence markups for each term it deems Comcog evidence, and the sentence span with the evidence in it (that has not been filtered out by non-relevant counter evidence) is used to create one or more of the Comcog sub category mark up. This is in addition to having the Comcog\_yes attributes set. There are two reasons for the creation of these additional markups:

* Visualization: in GATE and UIMA, you cannot see attributes. Making them markups with spans allow one to turn that layer on or off to see each of the attributes.
* Evaluation: The existing evaluation functionality counts at the mark-up level, not at the attribute level. It was easier to create markups than to alter the complicated evaluation code.

Each of the 17 of the 20 Comcog annotators work the same way. They look for terms that the lexicon has categorized with the Comcog subcategory within the span of the *Comcog\_yes* sentence. When found, a subcategory evidence markup is created and a subcategory markup is created for the span of the *Comcog\_yes*. Only one evidence or mention can be created for each possible subcategory.

Three of the four SSA mapping subcategories are treated differently. The methods that work on these subcategories are run after the other 17 have been run.

A prerequisite for the applied memory category being applied is that the Comcog\_yes span would also have to already been marked with a d163, d172, d177, d210, d230, d310, d330, or d350 mention before looking for any terms that would have an applied memory category from the lexicons.

Pacing and Persistence categories have the prerequisites that d130, d160, d160, d166, d170, d172, d175, d179, or d240 mentions within the span of the Comcog\_yes span would have to be present before looking for any terms that have to do with pacing or persistence categories from the lexicons.

# Efficacy

The efficacy for Comcog subcategories is below. The results for the training and validation are being provided to give an indication of the stability and how much degradation there was between training and testing sets. The intuition behind looking at the degradation also gives an impression of how similar the distribution of reportable classes between the training and testing sample. Given the small amount of overall data, it is nearly impossible to keep a balance between the sets for each subcategory.

*Table x*: Comcog Subcategory Efficacy At the Sentence Level, Testing Sample

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **F-Score** | **Recall** | **Precision** | **Total** | **TP** | **FP** | **FN** | **Notes** |
| **d110\_d129** | NaN | 0.000 | 0.000 | 2 | 0 | 0 | 2 | [1] |
| **d130\_d159** | NaN | 0.000 | 0.000 | 0 | 0 | 1 | 0 | [2] |
| **d160** | 0.706 | 1.000 | 0.545 | 6 | 6 | 5 | 0 | [3] |
| **d163** | 0.513 | 0.500 | 0.526 | 40 | 20 | 18 | 20 |  |
| **d166** | 0.400 | 0.500 | 0.333 | 2 | 1 | 2 | 1 | [4] |
| **d170** | NaN | 0.000 | 0.000 | 2 | 0 | 1 | 2 | [5] |
| **d172** | NaN | 0.000 | 0.000 | 0 | 0 | 1 | 0 | [2] |
| **d175** | 0.500 | 1.000 | 0.333 | 1 | 1 | 2 | 0 | [6] |
| **d177** | 0.714 | 0.694 | 0.735 | 36 | 25 | 9 | 11 |  |
| **d179** | NaN | 0.000 | 0.000 | 1 | 0 | 0 | 1 | [7] |
| **d210\_d220** | 0.391 | 0.321 | 0.500 | 56 | 18 | 18 | 38 | [8] |
| **d230** | NaN | 0.000 | 0.000 | 0 | 0 | 1 | 0 | [2] |
| **d240** | NaN | 0.000 | 0.000 | 1 | 0 | 1 | 1 | [9] |
| **d310\_d329** | 0.548 | 0.426 | 0.767 | 54 | 23 | 7 | 31 | [10] |
| **d330\_d349** | 0.755 | 0.693 | 0.830 | 176 | 122 | 25 | 54 | [11] |
| **d350\_d369** | 0.613 | 0.792 | 0.500 | 24 | 19 | 19 | 5 | [12] |
| **Applied Memory** | 0.495 | 0.529 | 0.466 | 51 | 27 | 31 | 24 | [13] |
| **Adaptation** | NaN | 0.000 | 0.000 | 0 | 0 | 14 | 0 | [1,2] |
| **Pacing** | NaN | 0.000 | 0.000 | 1 | 0 | 0 | 1 | [14] |
| **Persistence** | 0.267 | 0.400 | 0.200 | 5 | 2 | 8 | 3 | [15] |
| **Totals** | 0.597 | 0.576 | 0.618 | 458 | 264 | 163 | 194 |  |
| **Avg** | 0.537 | 0.623 | 0.521 |  |  |  |  | [16] |

Number Note Detail

1. No positive mentions in training set
2. No positive mentions in validation set
3. Stable performance across training(f=.74) to validation (f=.70)
4. Big drop off from training (f=.87) to validation (f=.39)
5. Only two positive mentions in validation
6. Training had plenty of examples; validation only had 1; which was gotten; but many validation fp's
7. Only one positive example in validation
8. d-210-d230 Undertaking tasks was a tough genre to do via terminology approach. Also; didn't do much tuning on it yet
9. Only 1 positive mention in validation set.
10. Big drop off from training (f=.77) to validation (f=.55)
11. Performance went up from training (f=.72) to validation (f=.75) - I'll call this category stable
12. Stable performance across training (f=.61) to validation (f=.61)
13. While the notion of Applied Memory is evolving from discussions around the fp's; it's performance looks stable with training (f=.51) to validation (f=.50)
14. Only one positive case in training and one positive case in validation
15. Big drop off in performance around Persistence with training (f=.69) to validation (f=.27)
16. The denominator was 11 rather than 20, because 9 of 20 were not valid runs for one of the above reasons

Table x: Comcog Subcategory Efficacy At the Sentence Level, Training Sample

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **F-Score** | **Recall** | **Precision** | **Total** | **TP** | **FP** | **FN** | **Notes** |
| **d110\_d129** | NaN | 0 | 0 | 0 | 0 | 2 | 0 | [1] |
| **d130\_d159** | 0.720 | 0.900 | 0.600 | 10 | 9 | 6 | 1 | [2] |
| **d160** | 0.788 | 0.867 | 0.722 | 15 | 13 | 5 | 2 | [3] |
| **d163** | 0.602 | 0.667 | 0.549 | 117 | 78 | 64 | 39 | [3.1] |
| **d166** | 0.824 | 0.875 | 0.778 | 8 | 7 | 2 | 1 | [3.2] |
| **d170** | 0.800 | 1.000 | 0.667 | 4 | 4 | 2 | 0 | [4] |
| **d172** | 0.667 | 0.833 | 0.556 | 6 | 5 | 4 | 1 | [5] |
| **d175** | 0.653 | 0.889 | 0.516 | 18 | 16 | 15 | 2 | [6] |
| **d177** | 0.788 | 0.860 | 0.727 | 136 | 117 | 44 | 19 |  |
| **d179** | 0.737 | 0.875 | 0.636 | 8 | 7 | 4 | 1 | [7] |
| **d210\_d220** | 0.693 | 0.758 | 0.639 | 182 | 138 | 78 | 44 | [8.1] |
| **d230** | 0.571 | 0.750 | 0.462 | 8 | 6 | 7 | 2 | [2] |
| **d240** | 0.737 | 0.700 | 0.778 | 10 | 7 | 2 | 3 | [9] |
| **d310\_d329** | 0.756 | 0.684 | 0.844 | 190 | 130 | 24 | 60 | [10] |
| **d330\_d349** | 0.733 | 0.706 | 0.762 | 517 | 365 | 114 | 152 | [11] |
| **d350\_d369** | 0.622 | 0.822 | 0.500 | 101 | 83 | 83 | 18 | [12] |
| **Applied Memory** | 0.517 | 0.634 | 0.437 | 153 | 97 | 125 | 56 | [13] |
| **Adaptation** | NaN | 0.000 | 0.000 |  | 0 | 46 | 0 | [1,2] |
| **Pacing** | 1.000 | 1.000 | 1.000 | 1 | 1 | 0 | 0 | [14] |
| **Persistence** | 0.687 | 0.687 | 0.687 | 16 | 11 | 5 | 5 | [15] |
| **Totals** | 0.678 | 0.729 | 0.634 | 1500 | 1094 | 632 | 406 |  |
| **Macro Avg** | 0.716 | 0.806 | 0.659 |  |  |  |  | [16] |

1. No positive mentions in training set

2. No positive mentions in validation set

3. Performance dropped some across training(f=.78) to validation (f=.70)

3.1. Performance dropped some across training(f=.60) to validation (f=.52)

3.2. Performance dropped some across training(f=.82) to validation (f=.66)

4. Big drop off from training (f=.87) to validation (f=.39)

5. Only two positive mentions in validation, and four positive mentions in training

6. Training had plenty of examples, validation only had 1, which was gotten, but many validation fp's

7. Only one positive example in validation

8.1. Some tuning done; some performance drop from training (f=.69) to validation (f=.64)

9. Only 1 positive mention in validation set.

10. Big drop off from training (f=.77) to validation (f=.55)

11. Performance went up from training (f=.73) to validation (f=.78) - I'll call this category stable

12. Stable performance across training (f=.62) to validation (f=.62)

13. While the notion of Applied Memory is evolving from discussions around the fp's , it's performance looks stable with training (f=.51) to validation (f=.49)

14. Only one positive case in training and one positive case in validation

15. Big drop off in performance around Persistence with training (f=.69) to validation (f=.28)

16. The Delimiter was 18 – accounting for the number of invalid runs described above.

# Failure Analysis

Noted: a small but obvious portion of the false negatives were from *Slot:Values* where the answer part to the slot or question was in a second segment. Answers, lacking the full context, were universally not marked. This was an engineering trade-off, to write-off those failures knowing they could be addressed, because the gain was small. I never got the cycles to write a wrapper to check whether a \_yes sentence is within a *Slot:Value*.

Noted: only a small number of false negatives were the result of out-of-terminology evidences. Out-of-terminology kinds of failures are easy to remedy when seen. It’s a question of training exposure to pick these up.

Noted: A significant number of false negatives were the result of non-obvious inferences. When there is time, this class of failures should be followed up in a feed-back loop with the annotators to verify these were truly positive cases, and to gather insight how to identify the context around these cases. For these classes of failure, this simplistic term based lookup technique may not be appropriate.

# Discussion

The training annotations were skewed to majority class with little to no coverage for attributes that would be most helpful for the SSA use case. In particular, *adaptation*, *persistence*, *pacing*, and *d110\_d129\_Learning and Applying Knowledge*. As such, the efficacy of these attributes are weak because the cell sizes are too low. We did gather terminology for these classes, ergo in theory, a follow-up would be to augment the existing samples with pages pulled using terms we know to be from these classes.