

# OHDSI NLP working group

## Implementation of term\_exists in NLP tools



MEDICAL UNIVERSITY  
of SOUTH CAROLINA

Changing What's Possible

Stephane M. Meystre, MD, PhD, FACMI

May 9, 2018

# Definition of term\_exists

## From the OMOP CDM v5.2 NOTE\_NLP table definition:

**Term\_exists** is defined as a flag that indicates if the patient actually has or had the condition.

Any of the following modifiers would make **Term\_exists false**:

Negation = true

Subject = [anything other than the patient]

Conditional = true

Rule\_out = true

Uncertain = very low certainty or any lower certainties

A complete lack of modifiers would make **Term\_exists true**. For the modifiers that are there, they would have to have these values:

Negation = false

Subject = patient

Conditional = false

Rule\_out = false

Uncertain = true or high or moderate or even low (could argue about low)



# CLAMP implementation of term\_exists

In CLAMP, **term\_exists** could be interpreted as follows:

- **Term\_exists = false** if :
  - Concept is negated (Assertion = absent); or
  - Concept has condition modifier (Attribute CON); or
  - Concept has subject modifier and the subject is not patient (Attribute SUB  $\neq$  patient or pt)
- **Term\_exists = true** otherwise,

**NB:** uncertainty not assessed

<https://clamp.uth.edu>

# MetaMap implementation of term\_exists

In general **term\_exists** could be interpreted as follows:

- **Term\_exists = false** if CUI tagged as negated:
  - Negation Flag = 1 in MMI output; or
  - CUI in NegConcepts/NegConCUI in XML or JSON outputs
- **Term\_exists = true** otherwise,

**NB:** subject, conditional, and uncertainty not assessed!

<https://metamap.nlm.nih.gov>





# MetaMap implementation of term\_exists

In **XML** output:

<b>&lt;MMOs&gt;</b> <b>&lt;MMO&gt;</b>	All the XML output generated for an entire input record or citation, consisting of <ul style="list-style-type: none"> <li>• <b>&lt;CmdLine&gt;</b>: the command used to start MetaMap,</li> <li>• <b>&lt;AA&gt;</b>: any acronyms/abbreviation(s) found in the text,</li> <li>• <b>&lt;Negation&gt;</b>: any negation(s) found in the text, and</li> <li>• <b>&lt;Utterances&gt;</b>: the utterance(s) found in the text</li> </ul>
<b>&lt;Negations Count="N"&gt;</b> <b>&lt;Negation&gt;</b>	All the data generated for a negation, including <ul style="list-style-type: none"> <li>• <b>&lt;NegType&gt;</b>: the negation type,</li> <li>• <b>&lt;NegTrigger&gt;</b>: the negation trigger,</li> <li>• <b>&lt;NegTriggerPI&gt;</b>: the negation trigger's positional information,</li> <li>• <b>&lt;NegConcepts&gt;</b>: the negated concept(s), and</li> <li>• <b>&lt;NegConcPIs&gt;</b>: the negated concept's StartPos/Length positional information</li> </ul>
<b>&lt;NegConcCUI&gt;</b>	The CUI associated with the negated concept
<b>&lt;NegConcepts Count="N"&gt;</b> <b>&lt;NegConcept&gt;</b>	The negated concept(s), consisting of <ul style="list-style-type: none"> <li>• <b>&lt;NegConcCUI&gt;</b>: the negated concept's CUI, and</li> <li>• <b>&lt;NegConcMatched&gt;</b>: the negated concept's name</li> </ul>
<b>&lt;NegConcMatched&gt;</b>	The name of the negated concept
<b>&lt;NegConcPIs Count="N"&gt;</b> <b>&lt;NegConcPI&gt;</b>	The StartPos/Length positional information of the negated concept
<b>&lt;NegTrigger&gt;</b>	The negation trigger
<b>&lt;NegTriggerPIs Count="N"&gt;</b> <b>&lt;NegTriggerPI&gt;</b>	The StartPos/Length positional information of the negation trigger
<b>&lt;NegType&gt;</b>	The negation type



# MetaMap implementation of term\_exists

In **JSON** output (Negations):

## 2.3 The Negations Pair

The **Negations** pair spans lines 39–58 in Appendix A, and represents the negations that MetaMap detected in the text. MetaMap's negation detection is based on Wendy Chapman's [NegEx](#), and includes some significant extensions, documented [here](#).

The only negated concept in the text is **heart attack**, which is negated by the trigger **no**; the data describing this AA are rendered in JSON as

```
{
  "NegType": "nega",
  "NegTrigger": "no",
  "NegTriggerPIs": [
    {
      "StartPos": "0",
      "Length": "2"
    }
  ],
  "NegConcepts": [
    {
      "NegConcCUI": "C0027051",
      "NegConcMatched": "-- Heart Attack"
    }
  ],
  "NegConcPIs": [
    {
      "StartPos": "3",
      "Length": "12"
    }
  ]
}
```

*Type of negation*  
*Negation trigger*  
*Negation trigger character offsets*  
*Negation trigger starting offset*  
*Negation trigger character length*  
*Negated concept(s)*  
*Negated concept's CUI*  
*Negated concept's string*  
*Negated concept character offsets*  
*Negated concept starting offset*  
*Negated concept character length*



# cTAKES implementation of term\_exists

In cTAKES 4.0, **term\_exists** could be interpreted as follows:

- **Term\_exists = false** if EventMention/EntityMention attribute(s)
  - polarity=-1 OR
  - subject="other" OR
  - uncertainty=1
- **Term\_exists = true** otherwise,

**NB:** conditional not assessed (rule\_out maybe not assessed by Assertion module)

<http://ctakes.apache.org>

<https://cwiki.apache.org/confluence/display/CTAKES/cTAKES>



# cTAKES implementation of term\_exists

Local context analysis options in cTAKES:

- **Assertion** component contains the `PolarityClearTkAnalysisEngine`, which is machine learning based.
- **NE Contexts** component contains the original negation annotator, which is implemented using the `ContextAnnotator`, rule-based.
- **NegexAnnotator** component in YTEX-UIMA, a drop-in replacement for the original cTAKES negation annotator.





# cTAKES implementation of term\_exists

cTAKES Assertion component assesses:

- **Negation** attribute annotator: polarity (1 or -1, -1=negated) `org.apache.ctakes.typesystem.type.constants.CONST.T.NE_POLARITY_NEGATION_ABSENT` and `CONST.NE_POLARITY_NEGATION_PRESENT`.
- **Uncertainty** attribute annotator: uncertainty (0 or 1, 1 = uncertain) `CONST.NE_UNCERTAINTY_PRESENT` and `CONST.NE_UNCERTAINTY_ABSENT`.
- **Generic** attribute annotator: generic (true or false). `CONST.NE_GENERIC_TRUE` and `CONST.NE_GENERIC_FALSE`
- **Subject** attribute annotator: subject ("patient", "other", and other values). Constants with the prefix `ATTR_SUBJECT_*` within class `org.apache.ctakes.typesystem.type.constants.CONST`



# NOBLE Coder implementation of term\_exists

In NOBLE, **term\_exists** could be interpreted as follows:

- **Term\_exists = false** if concept attribute(s) are:
  - Polarity=Negative\_Polarity OR
  - Experiencer≠Patient\_Experiencer OR
  - Certainty=DefiniteNegatedExistence\_Certainty
- **Term\_exists = true** otherwise,

**NB:** conditional not assessed (rule\_out maybe assessed with polarity)

<http://noble-tools.dbmi.pitt.edu>

# Summary of implementation of term\_exists

**term\_exists = false** if:

	OMOP CDM NOTE_NLP	CLAMP	MetaMap	cTAKES	NOBLE
Negation	true	true	true	true	true
Subject	<b>Not</b> patient	<b>Not</b> patient	-	<b>Not</b> patient	<b>Not</b> patient
Conditional	true	true	-	-	-
Rule-out	true	(negation true?)	negation =true	negation =true	(negation true?)
Uncertain	true (low certainty)	-	-	true	true (negated)



# Summary of implementation of term\_exists

**term\_exists = true** if:

Caveat: Not all attributes assessed,  
and term\_exists=true not certain

	OMOP CDM NOTE_NLP	CLAMP	MetaMap	cTAKES	NOBLE
Negation	false	false	false	false	false
Subject	patient	patient	-	patient	patient
Conditional	false	false	-	-	-
Rule-out	false	(negation false?)	negation =false	negation =false	(negation false?)
Uncertain	false (high, moderate certainty)	-	-	false	false (definite, probable)

