Practices in Text Annotation

Keigh Rim

1/31/2017

Good Old Text Annotation
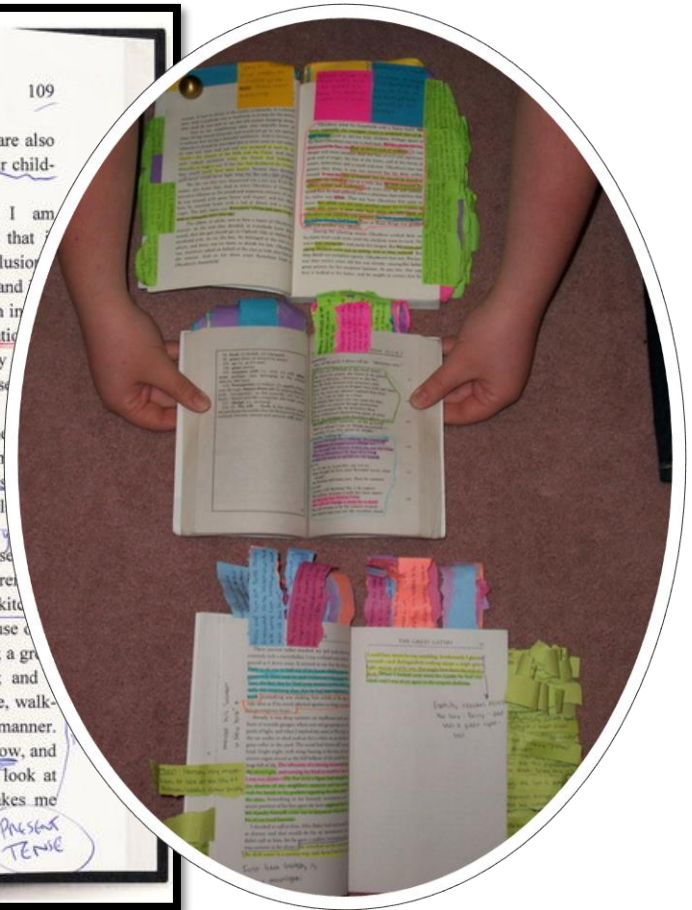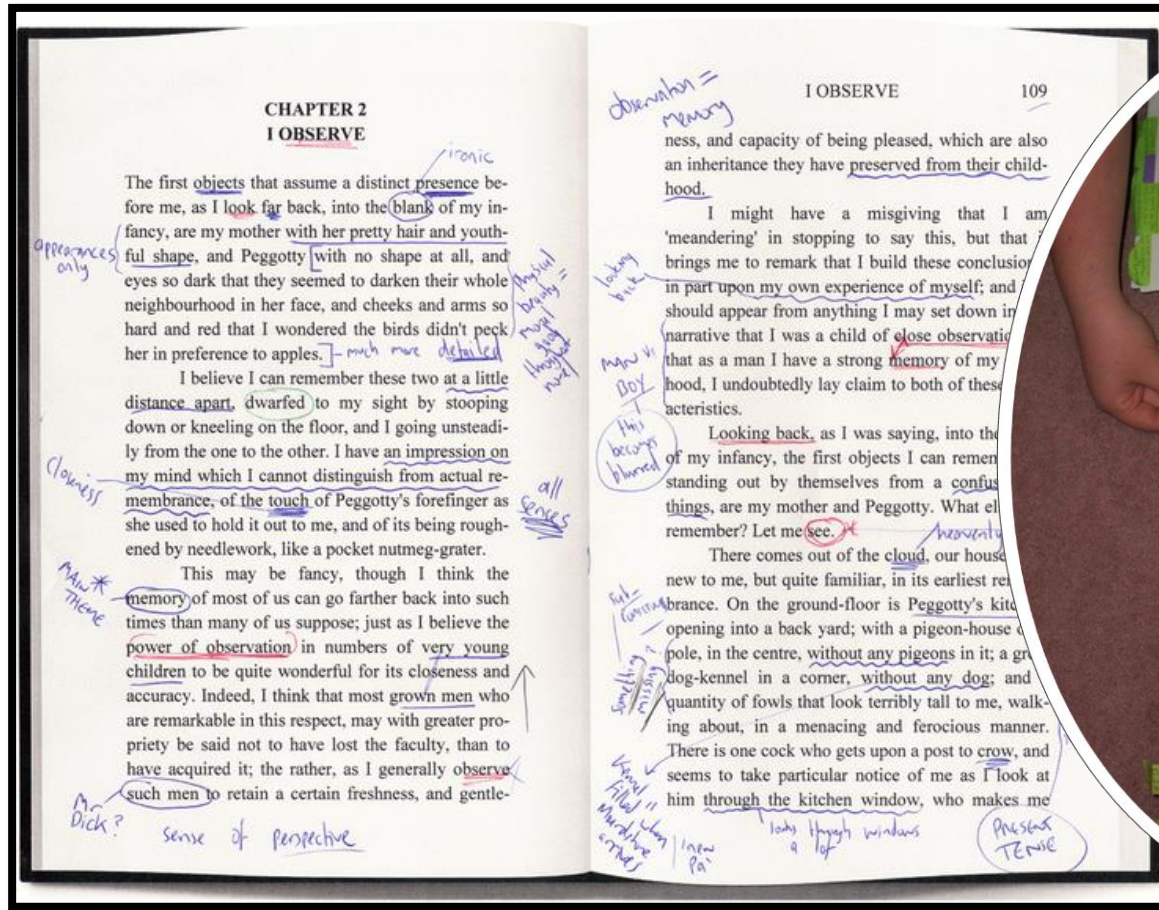
Modern World Annotation

Annotation as Language Resource

- Language Resources for NLP/ML should be ***Machine-readable***
  - Using common character set(s)
  - Serializable
  - Parse-compatible
  - Interoperable
- See Linguistic Annotation Framework (ISO standard #24612) literatures
  - Ide, N., & Romary, L. (2006). Representing Linguistic Corpora and Their Annotations.
  - Ide, N., & Suderman, K. (2007). GrAF: A Graph-based Format for Linguistic Annotations.

Scopes of Annotation Tasks

- Segmentation task
  - Entity identification
  - Find anchors, boundaries, or units
  - Also called unitization
  - *Can be non-consuming!*

- Linking task
  - Mark-up relations between entities/units
  - Only practicable once segmentation is given

- Labeling task
  - Giving names and attributes to segment and links

Column based data format

- MUC, CoNLL

	A	B
634	American	B-ORG
635	Depository	I-ORG
636	Receipts	I-ORG
637	(	O
638	ADR's	B-ORG
639	)	O
640	.	O
641		
642	Sólo	O
643	falta	O
644	el	O
645	visto	O
646	bueno	O
647	de	O
648	las	O
649	autoridades	O
650	brasileñas	O
651	para	O
652	hacer	O
653	efectiva	O
654	la	O
655	operación	O
656	de	O
657	compra	O
658	en	O
659	junio	O
660	próximo	O
661	.	O
662		
663	O	O
664	,	O
665	afirmó	O
666	.	O
667		
668	Por	O
669	Juan	B-PER
670	Miguel	I-PER
671	Núñez	I-PER

Development of XML

- Early XML serialization

In Washington **<TIMEX3 tid="t1" TYPE="DATE" VAL="PRESENT_REF" temporalFunction="true"
valueFromFunction="tf1" anchorTimeID="t0">**today**</TIMEX3>**, the Federal Aviation
Administration **<EVENT eid="e1" class="OCCURRENCE">**released**</EVENT>** air traffic control
tapes from the night the TWA Flight eight hundred
<EVENT eid="e2" class="OCCURRENCE">went**</EVENT>** down. There's nothing new on why
the plane **<EVENT eid="e3" class="OCCURRENCE">**exploded**</EVENT>**, but you
<EVENT eid="e4" class="OCCURRENCE">cannot**</EVENT>**
<EVENT eid="e5" class="OCCURRENCE">miss**</EVENT>** the moment.
ABC's Lisa Stark **<EVENT eid="e6" class="OCCURRENCE">**has**</EVENT>** more.

MARKABLES

<MAKEINSTANCE eventID="e1" pos="VERB" eiid="ei1" tense="PAST" aspect="NONE"/>
<MAKEINSTANCE eventID="e2" pos="VERB" eiid="ei2" tense="PAST" aspect="NONE"/>
<MAKEINSTANCE eventID="e3" pos="VERB" eiid="ei3" tense="PAST" aspect="NONE"/>
<MAKEINSTANCE eventID="e4" pos="VERB" eiid="ei4" tense="PRESENT" aspect="NONE"/>
<MAKEINSTANCE eventID="e5" pos="VERB" eiid="ei5" tense="INFINITIVE" aspect="NONE"/>
<MAKEINSTANCE eventID="e6" pos="NONE" eiid="ei6" tense="PRESENT" aspect="NONE"/>

NON-MARKABLES (EMPTY TAGS)

<TLINK eventInstanceID="ei1" relatedTo...="t1" relType="IS_INCLUDED" rule="2-1" />
<TLINK eventInstanceID="ei2" relatedTo...="..." relType="IS_INCLUDED" rule="2-1" />
<TLINK eventInstanceID="ei1" relatedToEventInst...="ei3" relType="BEFORE" rule="3-19" />
<TLINK eventInstanceID="ei3" relatedToEventInstance="ei4" relType="BEFORE" rule="6-1" />
<TLINK eventInstanceID="ei3" relatedToEventInstance="ei6" relType="BEFORE" rule="3-23" />

Stand-off Annotation

- Modern XML serialization in MAE

```
▼<NounVerbTask>
  ▼<TEXT>
    ▼<![CDATA[
        JABBERWOCKY By Lewis Carroll 'Twas brillig, and the slithy toves Did gyre and gimble in the wabe; All
        mimsy were the borogoves, And the mome raths outgrabe. 'Beware the Jabberwock, my son! The jaws that
        bite, the claws that catch! Beware the Jubjub bird, and shun The frumious Bandersnatch!' He took his
        vorpal sword in hand: Long time the manxome foe he sought-- So rested he by the Tumtum tree, And stood
        awhile in thought. And as in uffish thought he stood, The Jabberwock, with eyes of flame, Came whiffling
        through the tulgey wood, And burbled as it came! One, two! One, two! And through and through The vorpal
        blade went snicker-snack! He left it dead, and with its head He went galumphing back. 'And hast thou
        slain the Jabberwock? Come to my arms, my beamish boy! O frabjous day! Callooh! Callay!' He chortled in
        his joy. 'Twas brillig, and the slithy toves Did gyre and gimble in the wabe; All mimsy were the
        borogoves, And the mome raths outgrabe.
    ]]>
  </TEXT>
  ▼<TAGS>
    <NOUN id="N0" start="1" end="12" text="JABBERWOCKY" type="thing" comment=""/>
    <NOUN id="N1" start="61" end="66" text="toves" type="thing" comment="default value"/>
    <NOUN id="N2" start="94" end="98" text="wabe" type="place" comment="default value"/>
    <NOUN id="N3" start="119" end="128" text="borogoves" type="thing" comment="default value"/>
    <VERB id="V0" start="71" end="75" text="gyre" tense="past" aspect="simple"/>
    <VERB id="V1" start="80" end="86" text="gimble" tense="past" aspect=""/>
    <VERB id="V2" start="956" end="964" text="outgrabe" tense="" aspect="perfect progressive"/>
    <ADJ_ADV id="A2" start="37" end="44" text="brillig" type=""/>
    <ADJ_ADV id="A3" start="54" end="60" text="slithy" type=""/>
    <ADJ_ADV id="A4" start="104" end="109" text="mimsy" type=""/>
    <ACTION id="A0" fromID="V1" fromText="gimble" toID="N3" toText="borogoves" relationship="performed_by"/>
    <ACTION id="A1" fromID="N0" fromText="JABBERWOCKY" toID="V0" toText="gyre" relationship="performs"/>
    <DESCRIPTION id="D0" fromID="A3" fromText="slithy" toID="N1" toText="toves" relationship="describes"/>
    <DESCRIPTION id="D1" fromID="A4" fromText="mimsy" toID="N3" toText="borogoves" relationship=""/>
  </TAGS>
</NounVerbTask>
```

Interoperable Annotation

- Gate XML(Gate)

- UIMA XML (Dkpro)

- LIF JSON-LD(Lappsgrid)

- RDF & OWL (Semantic web community & Europe)

- …

From model to specification

- A model is
    - Elements (terms)
    - Their relations
    - Interpretation of them
- A model is formalized into a specification
    - Technical descriptions of the model
- An annotator does their job based on guideline
    - Verbal descriptions of the model

Document Type Definition

- DTD is used to define XML scheme
  - Writing a DTD is re-writing your annotation scheme (model) for an XML based output
  - In MAE, we use a bit simplified DTD
  - For full details, see http://www.w3schools.com/xml/xml_dtd_intro.asp

DTD for MAE

- DTDs have 3 parts
  1. Task Name
  2. Elements
     - Extent Tags: unitization task
     - Link Tags: linking task
  3. Element Attributes: detailed labeling task

Task Name

- You can define your task name with a line such as the following:

```
<!ENTITY name "myTask">
```

Elements

- Extent Tags are defined with:

```
<!ELEMENT TagName ( #PCDATA ) >
```

- Link Tags are defined with:

```
<!ELEMENT LinkName EMPTY >
```

Attributes, pre-defined

- Extent Tags are automatically given the following attributes:
  - id
  - spans
  - text
- Link Tags are automatically given the following attributes
  - id
  - to-argument
  - from-argument

Defining an Attribute

- Full syntax for defining an attribute

```
<!ATTLIST
        __TAG_NAME__
        __ATT_NAME__
        __VALUE_TYPE__
        prefix
        __REQUIRE__
        __DEF_VALUE__>
```

Defining an Attribute

- *_TAG_NAME_* : the name of tag an attribute is associated

- *_ATT_NAME_* : the name of the attribute

- *_VALUE_TYPE_* : one of ID, CDATA, IDREF, or a list (X | Y | ...)

- prefix : only used to give argument name for a link tag

- *_REQUIRE_* : #REQUIRE or #IMPLIED

- *_DEF_VALUE_* : the default value, double-quoted

More on predefined attributes

- id Attribute
  - The id attribute is generated automatically
  - Values are also automatically assigned by MAE
  - An optional ID prefix can be specified as follows:
    - Unless you want to give a specific prefix, you don't technically need to put this line in your DTD

```
<!ATTLIST TagName id ID prefix="TN" #REQUIRED >
```

- spans Attribute
  - Only when you want to allow an extent tag to be ***non-consuming***, you make spans optional

```
<!ATTLIST Tag1 spans #IMPLIED >
```

More on predefined attributes

- argN Attribute
  - A link tag grants from and to as its default argument
  - To define custom argument structure, use argN

```
<!ELEMENT ARGUMENTS EMPTY >
<!ATTLIST ARGUMENTS arg0 IDREF prefix="agent" #REQUIRED>
<!ATTLIST ARGUMENTS arg1 IDREF prefix="patient" #REQUIRED>
<!ATTLIST ARGUMENTS arg2 IDREF prefix="predicate" #REQUIRED>
```

  - Use prefix and required-ness for finer definition
  - All argN attributes automatically gets additional argNText attributes to keep text segment of the arguments