NLP with Elixir

Say *whuuut*?

Who am I?

Robert Bates

Research Scientist
Design & Intelligence Lab
School of Interactive Computing @ Georgia Tech

rob@octobang.com

arpieb most anywhere it matters...

What is NLP?

NLP is...

“Natural language processing is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.”

-- *Wikipedia*

(Not to be confused with “Neuro-Linguistic Programming” which is a whole other critter...)

Where have you experienced it?

- Your favorite search engine
- Chatbots
- Content indexers (Solr, Elasticsearch, etc)
- Automated support systems
- **Interactive agents**
  - **Siri**
  - **Alexa**
  - **Google Now**

Why is it important?

- People think in their native language, not hex or l33t (ok, most people)
- Most content out there is still in non-computer-friendly formats
  - We're still a long way from a truly semantic web
  - Competing "standards" and incomplete implementations
  - Legacy content!
- It's a natural method of interaction/socialization for humans
  - Ask questions (learning)
  - Carry on conversations
  - Aggregate information

What tools are already out there?

Tools...

Parsers: tokenizers, taggers, chunkers

- Natural Language Toolkit (aka NLTK) - Python
- Stanford Parser - Java
- spaCy - Python
- SyntaxNet - Python
- *CoreNLP, OpenNLP, etc ad nauseum...*

Semantic framing

- WordNet (relationships between words)
- VerbNet (hierarchical relationships between verbs)
- PropBank (expands VerbNet)
- FrameNet (conceptual hierarchical framing)
- SemLink (ties ‘em all together, where possible)

Tokenizer, Taggers, and Chunkers...

- Tokenizer
  - Breaks out words, punctuation
  - Handles stemming (plural vs singular, suffixes, etc)
  - Normalizes encodings (sometimes)
- Part of speech (POS) tagger
  - Identifies parts of speech (nouns, verbs, prepositions, etc)
  - Sometimes lemmatization (e.g. “is, was, were” => “be”)
- Chunker
  - Attempts to group words into “chunks” (noun phrases, verb phrases, etc)

Tokenizer, Taggers, and Chunkers...

“The fat cat sat on the mat.”

Tokenizer, Taggers, and Chunkers...

“The fat cat sat on the mat.”

Tokenizer + Tagger:

Word(u'**The**/DT'), Word(u'**fat**/JJ'), Word(u'**cat**/NN'), Word(u'**sat**/VBD'),
Word(u'**on**/IN'), Word(u'**the**/DT'), Word(u'**mat**/NN'), Word(u'./.')

Tokenizer, Taggers, and Chunkers...

“The fat cat sat on the mat.”

Tokenizer + Tagger:

Word(u'**The**/DT'), Word(u'**fat**/JJ'), Word(u'**cat**/NN'), Word(u'**sat**/VBD'),
Word(u'**on**/IN'), Word(u'**the**/DT'), Word(u'**mat**/NN'), Word(u'./.')

Chunker:

Chunk('**The fat cat**/NP'), Chunk('**sat**/VP'),
Chunk('**on**/PP'), Chunk('**the mat**/NP')

Semantic framing

Provides resources (data sets, XML, misc relational data, etc) for relating parts of speech to identifiable “thematic roles” in phrase patterns.

“The fat cat sat on the mat.”

- What are the important parts of speech in the phrase?
- What do they mean?
- Semantic framing FTW

Semantic framing

“The fat cat sat on the mat.”

('The fat cat/NP') ('sat/VP') ('on/PP') ('the mat/NP')

Semantic framing

“The fat cat sat on the mat.”

('The fat cat/NP') ('sat/VP') ('on/PP') ('the mat/NP')

NP V PP.location

Example: "The dog flopped in the corner."

Syntax: **Agent** V {{+loc}} **Location**

-- VerbNet 3.1, class *assuming_position-50*

The Holy Grail of NLP: ASRL

Automatic Semantic Role Labeling (ASRL)

- Use highly-accurate tokenizer/POS tagger/chunker (typically ML- and/or statistically-driven solution based on a priori corpora)
- Use combination of various semantic framing systems
- Map semantic frames into conceptual frames
- Removes/minimizes need for manual annotation and classification, automatically disambiguates phrases. (This is the real magik!)

The Holy Grail of NLP: ASRL

Are we there yet?

The Holy Grail of NLP: ASRL

Are we there yet?

Nope.

(But we *are* getting closer every year…)

Why use Elixir?

Why Elixir?

The usual suspects:

- Scalable
- Concurrent
- Distributed
- Etc ad nauseum...

But *this* is where it *really* shines IMHO:

- Pattern matching
- Unicode + strings support
- Metaprogramming
- Pipelining (aka transformative systems)
- Stack is optimized for hiperf service applications

Why Elixir?

Imagine if you will…

“The fat cat sat on the mat.”

Why Elixir?

Imagine if you will…

```
“The fat cat sat on the mat.”
|> tokenize()     # split out sentences/words
|> tag()          # tag parts of speech
|> chunk()        # identify phrase chunks
|> map_roles()    # map to semantic frames, roles
```

Why Elixir?

Imagine if you will…

```
“The fat cat sat on the mat.”
|> tokenize()     # split out sentences/words
|> tag()          # tag parts of speech
|> chunk()        # identify phrase chunks
|> map_roles()    # map to semantic frames, roles
```

… yields a data structure like, oh, say...

```
{:assuming_position_50, %{
Agent: [{“The”, :DT}, {“fat”, :JJ}, {“cat”, :NN}],
V: [{“sat”, :V}],
Location: [{“on”, :IN}, {“the”, :DT}, {“mat”, :NN}]
}}
```

Why Elixir?

… which could be handled by…

```
defmodule HeyAlexa do
    use NLPFrameHandler

    @doc ~S"""
    Process conceptual frame.
    """
    def handle_frame({:assuming_position_50, %{Agent:
    agent, V: verb, Location: location}) do
        # TODO
    end

    # Trusty catchall...
    def handle_phrase(_) do
        "Say whuuut?"
    end
end
```

Next steps...

- NLP is ultimately a transformation pipeline
- Create a functional, pluggable pipeline for processing content
  - Take inspiration from Phoenix plug approach; transforms HTTP request to response
- Make the architecture use-agnostic
  - Don’t hardwire it for something like Yet Another Chatbot…
  - … Or Phoenix...

Next steps...

Native parser tools

- Tokenizer
- Tagger
- Chunker

Initially leverage open, well-supported PCFG grammars like Stanford's while allowing custom/domain-specific grammars…?

Eventually dive into native xNN/ML solutions for more advanced parsers

Next steps...

Semantic framing

- VerbNet (WIP, POC)
- WordNet
- PropBank
- FrameNet
- SemLink

Basically provide packages with lookup APIs into frames and maps...

Da goods!
(aka Demos!)

References!

Parsers, Services, Informational

- http://www.nltk.org/
- https://spacy.io/
- http://nlp.stanford.edu/
- https://hex.pm/packages/gc_nlp
- https://hex.pm/packages/textgain

Semantic + Translation Data Sources

- https://verbs.colorado.edu/~mpalmer/projects.html
- https://verbs.colorado.edu/
- https://framenet.icsi.berkeley.edu/fndrupal/
- https://wordnet.princeton.edu/