Stanford CoreNLP *

David McClosky, Mihai Surdeanu, Chris Manning and many, many others

4/22/2011

* Previously known as BaselineNLProcessor
Overview

- Part I: KBP task overview
- Part II: Stanford CoreNLP
- Part III: NFL Information Extraction
Knowledge Base Population (KBP)

- **KBP** is a **bake-off** (shared task) held yearly

- **Task:** Given an entity, fill in values for various slots

- **Entities** can be people or organizations

- **Slots** are prespecified

- **Provenance** (textual sources) must be provided
Knowledge Base Population (KBP)

James T. Kirk
From Wikipedia, the free encyclopedia

James Tiberius "Jim" Kirk is a fictional character in the Star Trek media franchise. Kirk was first played by William Shatner as the principal lead character in the original Star Trek series. Shatner voiced Kirk in the animated Star Trek series and appeared in the first seven Star Trek movies. Chris Pine portrayed a younger version of the character in the 2009 Star Trek film, with Jimmy Bennett playing Kirk as a child. Other actors have played the character in fan-created media, and the character has been the subject of multiple spoofs and satires. Kirk also appears in numerous books, comics, and video games. The character has been praised for his leadership traits, but also criticized for his relationships with women.

Depiction

James T. Kirk was born and raised in Riverside, Iowa in the year 2233.[1][2] Diane Carey's novel Best Destiny identifies Kirk's parents as George and Winona Kirk.[3] Best Destiny and Carey's Final Frontier novel describe George Kirk's adventures aboard the USS Enterprise under the command of Captain Robert April. Although born on Earth, Kirk for a time lived on Tarsus IV, where he was one of nine surviving witnesses to the massacre of 4,000 colonists by Kodos the Executioner (Arnold Moss).[4] James Kirk's brother George Samuel Kirk is first mentioned in "What Are Little Girls Made Of?" and introduced and killed in "Operation: Annihilation", leaving behind three children one of which is seen in the same episode in pain because of the same reason like his dad but his fate is left unknown.[5]

At Starfleet Academy, Kirk became the only student to defeat the Kobayashi Maru test, gaining a commendation for original thinking by reprogramming the computer to make the "no-win scenario" winnable.[6] Kirk was granted a field commission as an ensign and posted to advanced training aboard the USS Republic.[7] He then was promoted to lieutenant junior grade and returned to Starfleet Academy as a student instructor.[8] Students could either "sink or swim" in his class, and Kirk himself was "a stack of books with legs".[9] Upon graduating in the top five percent, Kirk was promoted to lieutenant and assigned aboard the USS Enterprise.[10] While assigned to the Enterprise, Kirk commanded his first Starfleet away mission and survived a
James T. Kirk was born and raised in Riverside, Iowa in the year 2233. Diane Carey's novel Best Destiny identifies Kirk's parents, George and Winona Kirk. Best Destiny and Carey's Final Frontier novel describe George Kirk's adventures aboard the USS Enterprise. Because of the same reason like his dad but his fate is left unknown.

At Starfleet Academy, Kirk became the only student to defeat the Kobayashi Maru test, garnering a commendation for original thinking by reprogramming the computer to make the
Knowledge Base Population (KBP)

James T. Kirk

James T. Kirk was born and raised in Riverside, Iowa in the year 2233.[1] Diane Carey's novel Best Destiny identifies Kirk's parents as George and Winona Kirk.[2] Best Destiny and Carey's Final Frontier novel describe George Kirk's adventures aboard the USS Enterprise under the command of Captain Robert April. Although born on Earth, Kirk for a time lived on Tarsus IV, which is mentioned in "What Are Little Girls Made Of?" and introduced and killed in "Operation: Annihilate!", leaving behind three children one of which is seen in the same episode in pain because of the same reason like his dad but his fate is left unknown.[1]

At Starfleet Academy, Kirk became the only student to defeat the Kobayashi Maru test, garnering a commendation for original thinking by reprogramming the computer to make the
James T. Kirk was born and raised in Riverside, Iowa in the year 2233.\[1\] Diane Carey's novel *Best Destiny* identifies Kirk's parents as George and Winona Kirk.\[2\] *Best Destiny* and Carey's *Final Frontier* novel describe George Kirk's adventures aboard the USS *Enterprise* under the command of Captain Robert April. Although born on Earth, Kirk for a time lived on Tarsus IV, where he was one of nine surviving witnesses to the massacre of 4,000 colonists.

*Annihilate*\[3\] leaving behind three children one of which is seen in the same episode in pain because of the same reason like his dad but his fate is left unknown.\[1\]

At Starfleet Academy, Kirk became the only student to defeat the *Kobayashi Maru* test, garnering a commendation for original thinking by reprogramming the computer to make the...
Task Specification

• Inputs
  • Knowledge base
    • Entities, slot names and fillers
  • Source collection
    • Wikipedia, newswire text, broadcast news
  • Evaluation data from 2009, 2010
    • Queries (entities)
    • Slot names and fillers
  • Testing: Queries

• Output
  • Slot name, filler, document ID (provenance)
KBP Challenges: Slot types

per:parents: George and Winona Kirk
per:city_of_birth: Riverside
per:stateorprovince_of_birth: Iowa
per:date_of_birth: 2233
per:title: Captain
per:parents: George and Winona Kirk

per:parents: George Kirk

per:parents: Winona Kirk

mentioned in "What Are Little Girls Made Of?" and introduced and killed in "Operation: Annihilation", leaving behind three children one of which is seen in the same episode in pain because of the same reason like his dad but his fate is left unknown.[11]

At Starfleet Academy, Kirk became the only student to defeat the Kobayashi Maru test, garnering a commendation for original thinking by reprogramming the computer to make the "no-win scenario" winnable.[12] Kirk was granted a field commission as an ensign and posted to advanced training aboard the USS Republic.[13]

He then was promoted to lieutenant junior grade and returned to Starfleet Academy as a student instructor.[14] Students could either "sink or swim" in his class, and Kirk himself was "a stack of books with legs".[15] Upon graduating in the top five percent, Kirk was promoted to lieutenant and second aboard the USS Pegasus.[16] While assigned to the Pegasus, Kirk commanded his first planetary survey and survived a
James T. Kirk

per:city_of_birth: Shi‘Kahr

per:stateorprovince_of_birth: Iowa
James T. Kirk (Union officer)

James Thompson Kirk (September 21, 1820 – December 7, 1866) was a Pennsylvania merchant and then an officer in the Union Army during the American Civil War. He was the second officer to hold the title of colonel in the 10th Pennsylvania Reserve Regiment.[1]

Biography

Kirk was born in Caronosburg, Pennsylvania, a son of George A. and Jane (Thompson) Kirk. He was educated in the common schools and then became a merchant tailor. In 1851, he moved to Washington, Pennsylvania, where he owned and operated a retail store.[2] Kirk enlisted in the Army shortly after the outbreak of the Civil War and was commissioned in June 1861 as the captain of Company D of the 10th Pennsylvania Reserve Regiment (also known as the 39th Pennsylvania Infantry). Three weeks later, he became the regiment's lieutenant colonel.[3] He saw his first combat in the Battle of Dranesville in Northern Virginia, on December 20, 1861.

When Colonel John S. McClernond resigned in May 1862, Kirk was promoted to colonel. He commanded the regiment during the battles of Beaver Dam Creek, Gaines Mill, Glendale and Second Bull Run. In the first three actions, the regiment belonged to the 3rd Brigade of the 3rd Division of Maj. Gen. Fitz-John Porter's V Corps. At Second Bull Run, it fought with the 3rd Brigade of John F. Reynolds's independent division. He briefly led the brigade before being wounded and out of action. As a result of his injuries, he missed the battles of South Mountain and Antietam. When it became evident that his wounds would prevent any further field duty, he resigned from the army on October 18, 1862.[4]

Following the war, Kirk resumed his mercantile business. He died at the age of 60 in Washington and is buried in Washington Cemetery.[5]

References


Notes

3. 10th Pennsylvania Reserve website
4. Find-a-Grave

Categories: Union Army officers | People of Pennsylvania in the American Civil War | People from Washington County, Pennsylvania | Pennsylvania Reserves | 1826 births | 1866 deaths
Temporal Extensions

- Temporal slot filling:
  - `per:spouse`
  - `per:title`
  - `org:top_employees/members`
  - ...

- Simple representation:
  - \([T_1, T_2, T_3, T_4]\)
    - \(T_1 \leq \text{start} \leq T_2, \ T_3 \leq \text{end} \leq T_4\)
  - Any can be null to indicate a lack of constraint
  - Day resolution (YYYYMMDD)
Temporal Example

per:title: Captain
[22650101, 22651231, 22700101, 22701231]
Temporal Example

per:title: Captain
[22650101, 22651231, 22700101, 22701231]

Start is sometime during the year 2265
Overview

- Part I: KBP task overview
- Part II: Stanford CoreNLP
- Part III: NFL Information Extraction
We’ve been working on a whole bunch of stuff:

- Joint NLP models
- Coreference (now in Stanford CoreNLP)
- Supervised relation extraction (NFL)
- Supervised event extraction (BioNLP)
- Distantly supervised relation extraction (KBP)
- Scenario templates and graph models in IE

This section describes our NLP pipeline, common to many of these components.

http://nlp.stanford.edu/software/corenlp.shtml
Stanford CoreNLP Outline

• Approach
• How to use
  • Command-line (shell, batch)
  • Java interface
Motivation

- Quickly and painlessly get linguistic annotations for a text
- Hides variations across components behind common API
- Simple Java objects passed around (no XML, UIMA, etc.)
  - But results can easily be written to XML, etc.
Idea

Store the input text as well as the output of each Annotator as values in an Annotation Map.

- There are dependencies between Annotators ➔ the pipeline ordering is important!
Stanford CoreNLP Pipeline

- Tokenization (tokenize)
- Sentence Splitting (ssplit)
- Part-of-speech Tagging (pos)
- Morphological Analysis (lemma)
- Named Entity Recognition (ner)
- Syntactic Parsing (parse)
- Coreference Resolution (dcoref)
- NFL Relation Extraction (nfl)

Execution Flow

Raw text → Annotation Object → Annotated text
Stanford CoreNLP Outline

- Approach
- How to use
  - Command-line (shell, batch)
  - Java interface
Running as a shell


Example sentence: Stanford is located in California.

Sentence #1 (6 tokens):
- `[Word=Stanford Current=Stanford Tag=NNP Lemma=Stanford NER=ORGANIZATION]`
- `[Word=is Current=is Tag=VBZ Lemma=be NER=O]`
- `[Word=located Current=located Tag=VBN Lemma=locate NER=O]`
- `[Word=in Current=in Tag=IN Lemma=in NER=O]`
- `[Word=California Current=California Tag=NNP Lemma=California NER=LOCATION]`

(ROOT (S (NP (NNP Stanford))
  (VP (VBZ is) (VP (VBN located) (PP (IN in) (NP (NNP California))))))
  (. .)))

nsubjpass(located-3, Stanford-1)
auxpass(located-3, is-2)
prep_in(located-3, California-5)
Running in batch mode


Running in Java

Annotator pipeline =
    new StanfordCoreNLP(properties);

Annotation annotation =
    new Annotation(text);

pipeline.annotate(annotation);
Annotators in CoreNLP

- **tokenize** – split text into tokens, PTB-style
- **cleanxml** – remove specific XML tags
- **truecase** – restore case (e.g. if all lowercase, etc.)
- **ssplit** – sentence splitter
- **pos** – add POS tags to tokens
- **lemma** – add lemmas to tokens
More annotators in CoreNLP

- ner – add named entity tags to tokens
- regexner – add rule-based NER tags from regular expressions
- parse – add parse trees (Stanford Parser)
- berkeleyparse, charniakparse – Add parse trees from other parsers as well
- dcoref – add coreference links
- nfl (Machine Reading distribution only) – add NFL entity and relation extraction mentions
- time – add temporal annotations (coming later!)
Interpreting the Output

```java
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);
for (CoreMap sentence : sentences) {
    // traversing the words in the current sentence
    for (CoreLabel token: sentences.get(i).get(TokensAnnotation.class)) {
        String word = token.get(TextAnnotation.class);
        String pos = token.get(PartOfSpeechAnnotation.class);
        String ne = token.get(NamedEntityTagAnnotation.class);
    }
    // this is the parse tree of the current sentence
    Tree tree = sentence.get(TreeAnnotation.class);
}

// this is the coreference link graph
List<Pair<IntTuple, IntTuple>> graph = annotation.get(CorefGraphAnnotation.class);
```
Interpreting the Output

```java
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);
for (CoreMap sentence : sentences) {
    // traversing the words in the current sentence
    for (CoreLabel token: sentences.get(i).get(TokensAnnotation.class)) {
        String word = token.get(TextAnnotation.class);
        String pos = token.get(PartOfSpeechAnnotation.class);
        String ne = token.get(NamedEntityTagAnnotation.class);
    }
    // this is the parse tree of the current sentence
    Tree tree = sentence.get(TreeAnnotation.class);
}

// this is the coreference link graph
List<Pair<IntTuple, IntTuple>> graph = annotation.get(CorefGraphAnnotation.class);
```
Interpreting the Output

```java
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);
for (CoreMap sentence : sentences) {
    // traversing the words in the sentence
    for (CoreLabel token: sentences.get(i).get(TokensAnnotation.class)) {
        String word = token.get(TextAnnotation.class);
        String pos = token.get(PartOfSpeechAnnotation.class);
        String ne = token.get(NamedEntityTagAnnotation.class);
    }
    // this is the parse tree of the current sentence
    Tree tree = sentence.get(TreeAnnotation.class);
}

// this is the coreference link graph
List<Pair<IntTuple, IntTuple>> graph = annotation.get(CorefGraphAnnotation.class);
```
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);
for (CoreMap sentence : sentences) {
    // traversing the words in the sentence
    for (CoreLabel token : sentences.get(i).get(TokensAnnotation.class)) {
        String word = token.get(TextAnnotation.class);
        String pos = token.get(PartOfSpeechAnnotation.class);
        String ne = token.get(NamedEntityTagAnnotation.class);
    }
    // this is the parse tree of the current sentence
    Tree tree = sentence.get(TreeAnnotation.class);
}

// this is the coreference link graph
List<Pair<IntTuple, IntTuple>> graph = annotation.get(CorefGraphAnnotation.class);

hash map with class objects as keys and custom value types
CoreMap with additional properties (HasWord, HasTag, etc.)

uniquely identify a word by <sentence position, token position> (both offsets start at 0) [note: annotation will change soon…]
Creating Your Own Annotator

/** Simple annotator that recognizes locations stored in a gazetteer */
public class GazetteerLocationAnnotator implements Annotator {
    // this is the only method that must be implemented by an annotator
    public void annotate(Annotation annotation) {
        // traverse all sentences in this document (assumes that text already tokenized)
        for (CoreMap sentence : annotation.get(SentencesAnnotation.class)) {
            // loop over all tokens in sentence
            List<CoreLabel> tokens = sentence.get(TokensAnnotation.class);
            for (int start = 0; start < tokens.size(); start++) {
                // assumes that the gazetteer returns the token index
                // after the match or -1 otherwise
                int end = Gazetteer.isLocation(tokens, start);
                if (end > start) {
                    for (int i = start; i < end; i++) {
                        tokens.get(i).set(NamedEntityTagAnnotation.class, "LOCATION");
                    }
                }
            }
        }
    }
}
Overview

- Part I: KBP task overview
- Part II: Stanford CoreNLP
- Part III: NFL Information Extraction
NFL Information Extraction

- Add “nfl” to the “annotators” property.
- Construct and call the same way.
- Interpreting output:

```java
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);
for (CoreMap sentence : sentences) {
    List<EntityMention> entities = sentence.get(MachineReadingAnnotations.EntityMentionsAnnotation.class);
    List<RelationMention> relations = sentence.get(MachineReadingAnnotations.RelationMentionsAnnotation.class);
}
```
Useful methods

• **ExtractionObject:**
  • CoreMap getSentence()
  • Span getExtent()
  • String getType()
  • Counter<String> getTypeProbabilities()

• **EntityMention:**
  • int getSyntacticHeadTokenPosition()
  • CoreLabel getSyntacticHeadToken()

• **RelationMention:**
  • List<ExtractionObject> getArgs()
  • List<String> getArgNames()
NFL System Overview

- Named entity recognition
- Relation extraction

(Throughout: Lessons from adapting our IE system to NFL domain)
NFL System Overview

- Named entity recognition
- Relation extraction

(Throughout: Lessons from adapting our IE system to NFL domain)
CRF entity extractor

- Use NER system to classify each token as one of the NFL entity types or “o” (other)

- Contiguous tokens of the same type are combined into EntityMentions

- Marginal probabilities on each token form the results of getTypeProbabilities()
Extended the NFL team gazetteer with \texttt{NFLGame} entities extracted from Dekang Lin’s distributional similarity dictionary:

- seeds: win, loss, game
- added: victory, triumph, shutout, defeat, lead, match, rout, strikeout...

- If a word sequence (partially) matches a gazetteer entry and it includes the head of a NP $\Rightarrow$ gazetteer label
- If the generic NER labels a sequence as \texttt{DATE} $\Rightarrow$ Date
- If the generic NER labels a sequence as \texttt{NUMBER} and it is a valid score and not followed by measurement unit $\Rightarrow$ Score

Goal: maximize recall!
CRF + MTurk

- Harvested 1400+ sentences on NFL games from sports.yahoo.com
  - “It was the third quarter of the Philadelphia Eagles' 38-10 rout of the Carolina Panthers on Sunday and both franchises suddenly had big worries about their veteran quarterbacks.”
- Tagged corpus with rule-based NER, which maximized recall
- Generated MTurk HITs from this data, using all possible relations between the identified NEs
  - “Is it true that the Philadelphia Eagles scored 38 points in this game?” ➞ yes
  - “Is it true that the Philadelphia Eagles scored 10 points in this game?” ➞ no
- Averaged annotations from four annotators for each HIT
CRF + MTurk: Analysis

- MTurk helped only up to a point…
- Why?
  - There was a bug in the rule-based NER used to generate candidates
  - Turkers could not identify subtle mistakes, hence errors propagated in the final MTurk corpus
    - “… the victory game against Dallas”
    - Is “victory” the best word to describe the game? ➔ yes
    - Is “game” the best word to describe the game? ➔ yes
Adapting Entity Extraction to NFL

- Gazetteer used for team names, “game” entities
  - “Packers” should match “Green Bay Packers”
  - …but “Bay” shouldn’t.

- Tokenizer wasn’t splitting scores (“37-7”)

- Head finder needed adjustments
  - Heads of entities are critical features for both extraction tasks
NFL System Overview

- Named entity recognition
- Relation extraction

(Throughout: Lessons from adapting our IE system to NFL domain)
Relation Extraction Approach

- Logistic regression classifier
  - Positive datums: annotated relations in the corpus
  - Negative datums: all other possible combinations between existing entities
  - Example:
    - “It was the third quarter of the Philadelphia Eagles' 38-10 rout of the Carolina Panthers on Sunday and both franchises suddenly had big worries about their veteran quarterbacks.”
    - Positive: `teamScoringAll("Philadelphia Eagles", 38)`
    - Negative: `teamScoringAll("Philadelphia Eagles", 10)`
  - Features:
    - info on the entities in the relation
    - syntactic path between entities (both dependencies and constituents)
    - surface path between entities
    - entities between the relation elements
Basic NFL Inference

- Relation classifier is one-against-many
  - Can only predict one relation per pair of entities

- NFL domain often violates this!
  - \texttt{gameWinner(team, game) \rightarrow teamInGame(team, game)}

- System also doesn’t understand domain semantics, e.g.:
  - Games have exactly one winner and one loser.
  - Teams with higher scores win.

- Simple logical rules fill in some of these cases.
## Extraction results

<table>
<thead>
<tr>
<th>System</th>
<th>Entity Mentions</th>
<th>Relation Mentions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>73.7</td>
<td>49.7</td>
</tr>
<tr>
<td>+ gazetteer features</td>
<td>74.0</td>
<td>50.2</td>
</tr>
<tr>
<td>+ rule-based model for NFLTeam</td>
<td>75.5</td>
<td>53.2</td>
</tr>
<tr>
<td>+ improved head finding</td>
<td>76.1</td>
<td>57.9</td>
</tr>
<tr>
<td>+ basic inference</td>
<td>76.1</td>
<td>59.5</td>
</tr>
</tbody>
</table>
Summary

- Work on TAC-KBP and MR-KBP

- Use Stanford CoreNLP! 😊
  http://nlp.stanford.edu/software/corenlp.shtml

- NFL system builds on top of CoreNLP
Thank you!

Questions?