Event Extraction Using Distant Supervision

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D. Manning, Daniel Jurafsky

30 May 2014

Language Resources and Evaluation Conference

Reykjavik, Iceland

Overview

- **Problem:** Information extraction systems require lots of training data. Human annotation is expensive and does not scale.

- **Distant supervision:** Generate training data automatically by aligning existing knowledge bases with text.

  - Approach shown for relation extraction: Minz et al. 2009 (ACL); Surdeanu et al. 2012 (EMNLP).

- **Goal:** Adapt distant supervision to event extraction.

Outline

- Present new dataset and extraction task.
- Describe distant supervision framework.
- Evaluate several models within this framework.

Plane Crash Dataset

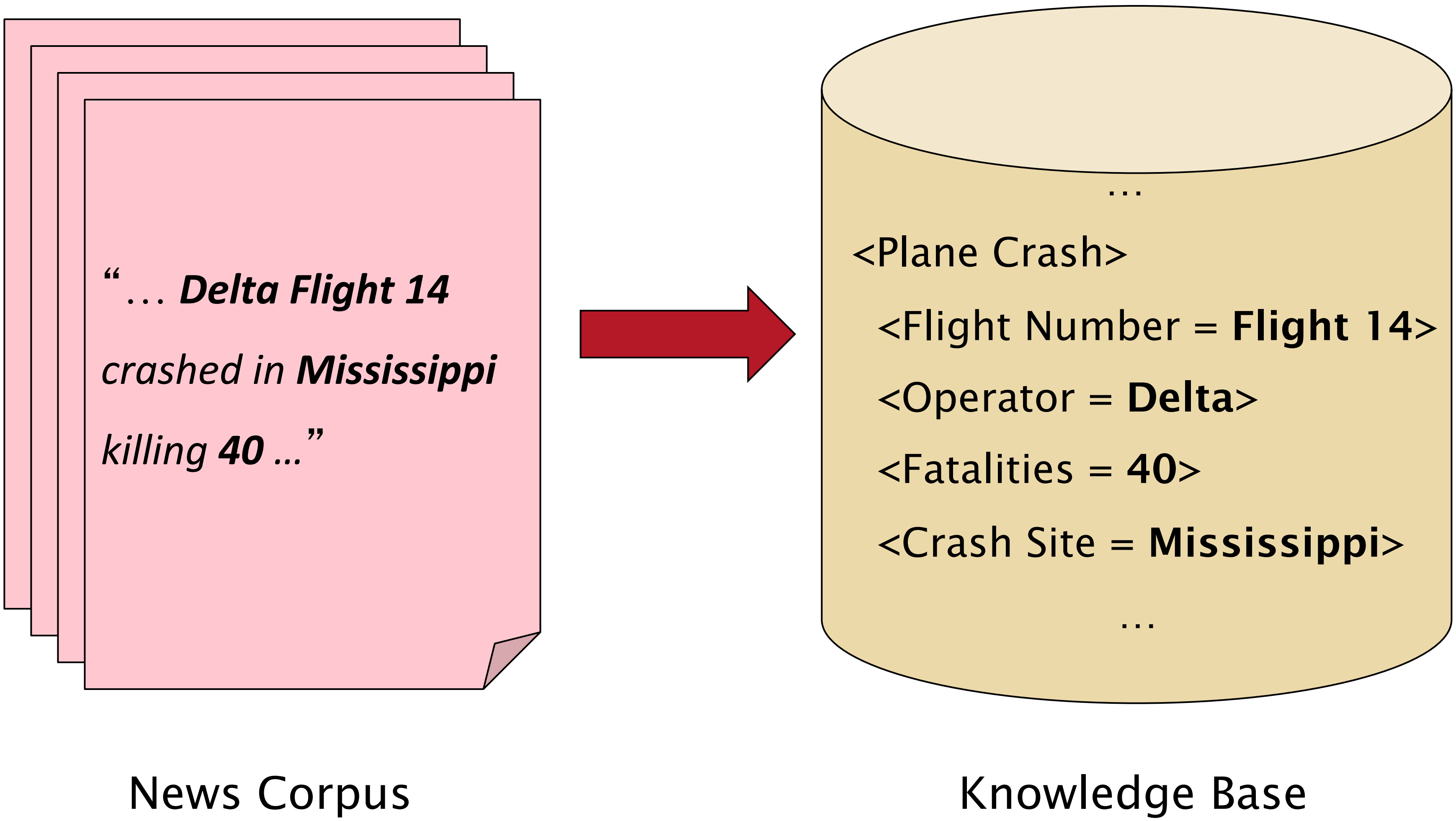
- 80 plane crash events from Wikipedia infoboxes (40 train / 40 test).
- Newswire corpus from 1988 to present (Tipster/Gigaword).
- Download: http://nlp.stanford.edu/projects/dist-sup-event-extraction.shtml

Comair Flight 3272

A Comair Embraer EMB-120, similar to the one involved.

Accident summary	
Date	January 9, 1997
Summary	Atmospheric icing leading to loss of control
Site	Monroe, Michigan, USA 41°57′48.08″N 83°33′8.39″W
Passengers	26
Crew	3
Fatalities	29 (all)
Survivors	0
Aircraft type	Embraer 120 RT Brasilia
Operator	Comair (as Delta Connection)
Registration	N265CA

Template-Based Event Extraction



Distant Supervision (Relation Extraction)

- **Noisy Labeling Rule:** If slot value and entity name appear together in a sentence, then assume that sentence encodes the relation.

Training Fact:
Entity: Apple
founder = Steve Jobs

Apple *co-founder* ***Steve Jobs*** *passed away in 2011.*
founder

Noise!!!

Steve Jobs *was fired from* ***Apple*** *in 1985.*
founder

Distant Supervision (Event Extraction)

- Sentence level labeling rule won't work.
  1. Many events lack proper names.
     - "The crash of USAir Flight 11"
  2. Slots values occur separate from names.
     - *The plane went down in central* ***Texas***.
     - ***10*** *died and* ***30*** *were injured in yesterday's tragic incident.*
- Heuristic solution:
  - Document-level labeling rule.
  - Use Flight Number as proxy for event name.

Training Fact:
{<Flight Number = Flight 11>,
<CrashSite= Toronto>}

*…**Flight 11** crash Sunday…*
…The plane went down in
[***Toronto***]$_{\text{CrashSite}}$…

Automatic Labeling Results

- 38,000 Training Instances.

Label	Frequency	Named Entity Type
NIL	19196	
Crash Site	10365	LOCATION
Operator	4869	ORGANIZATION
Fatalities	2241	NUMBER
Aircraft Type	1028	ORGANIZATION
Crew	470	NUMBER
Survivors	143	NUMBER
Passengers	121	NUMBER
Injuries	0	NUMBER

- 39% Noise:

Good: *At least 52 people survived the crash of the* ***Boeing 737***.

Bad: *First envisioned in 1964, the* ***Boeing 737*** *entered service in 1968.*

Model 1: Simple Local Classifier

- Multiclass Logistic Regression

- **Features:** unigrams, POS, NETypes, part of doc, dependencies

US Airways Flight 133 crashed in **Toronto**

LexIncEdge-prep_in-crash-VBD
UnLexIncEdge-prep_in-VBD
PREV_WORD-in
2ndPREV_WORD-crash
NEType-LOCATION
Sent-NEType-ORGANIZATION
etc.

Model 2: Sequence Model with Local Inference (SMLI)

- Intuition: There are dependencies between labels.

  Crew and Passenger go together:

  4 crew and **200** passengers were on board.

  Site often follows Site:

  The plane crash landed in **Beijing**, **China**.

  Fatalities never follows Fatalities

  * **20** died and **30** were killed in last Wednesday's crash.

- Solution: A sequence model where previous non-NIL label is a feature.
  - At train time: use noisy "gold" labels.
  - At test time: use classifier output.

Motivating Joint Inference

- Problem: Local sequence models propagate error.

20 dead, **15** injured in a **USAirways Boeing 747** crash.

Gold:	Fat.	Inj.	Oper.	A.Type.
Pred:	Fat.	Surv.	??	??

Motivating Joint Inference

- Problem: Local sequence models propagate error.

20 dead, **15** injured in a **USAirways Boeing 747** crash.

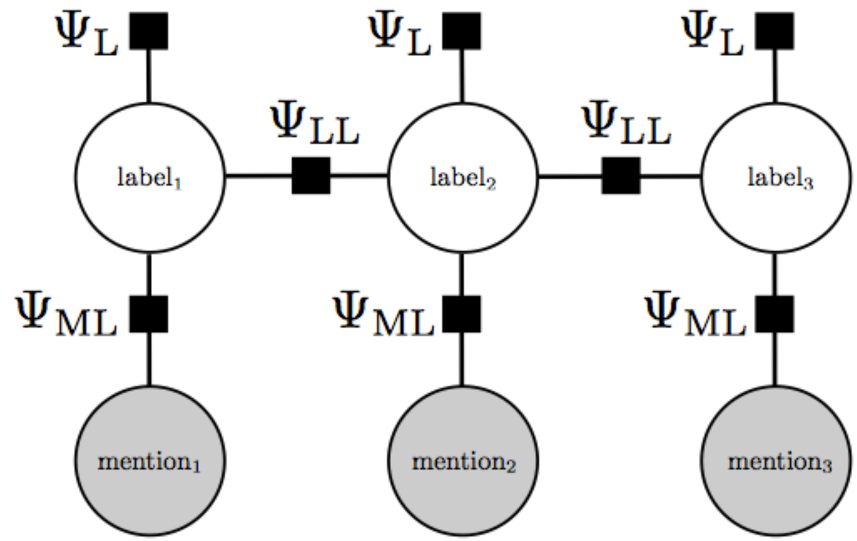
~~Gold: Fat. Inj. Oper. A.Type.~~

~~Pred: Fat. Surv. ?? ??~~

Gold:	Fat.	Fat.	Oper.	A.Type.
Pred:	Fat.	Inj.	??	??

Model 3: Condition Random Fields (CRF)

- Linear-chain CRF.
  - Algorithm: Laferty et al. (2001).
  - Software: *Factorie*. McCallum et al. (2009)



- Jointly model all entity mentions in a sentence.

Model 4: Search-based structured prediction (Searn)

- General framework for infusing global decisions into a structured prediction task (Daumé III, 2009).

- We use Searn to implement a sequence tagger over a sentence's entity mentions.

- Searn's "chicken and egg" problem:
  - Want to train an optimal classifier based on a set of global costs.
  - Want global costs to be computed from the decisions made by an optimal classifier.
  - Solution: Iterate!

A Searn iteration

- Start with classifier H_i.
- For each training mention:
  - Try all possible labels.
  - Based on label choice, predict remaining labels using H_i.
  - Compute global cost for each choice.

	20	dead, 15	injured in a USAirways	Boeing 747 crash.
Gold:	Fat.	Fat.	Oper.	A.Type
H_i:	Fat.			

- Use computed costs to train classifier H_{i+1}.

A Searn iteration

- Start with classifier H_i.
- For each training mention:
  - **Try all possible labels.**
  - Based on label choice, predict remaining labels using H_i.
  - Compute global cost for each choice.

	20 dead,	**15** injured in a	**USAirways**	**Boeing 747** crash.
Gold:	Fat.	Fat.	Oper.	A.Type
H_i:	Fat. →	Fat. Inj. etc…		

- Use computed costs to train classifier H_{i+1}.

A Searn iteration

- Start with classifier H_i.
- For each training mention:
  - Try all possible labels.
  - **<u>Based on label choice, predict remaining labels using H_i.</u>**
  - Compute global cost for each choice.

20 dead, <u>**15**</u> injured in a **USAirways Boeing 747** crash.

<u>Gold</u>: Fat. Fat. Oper. A.Type

H_i: Fat. → Fat. → NIL → NIL

Fat. → Inj. → Oper. → A. Type

Fat. → etc…

- Use computed costs to train classifier H_{i+1}.

A Searn iteration

- Start with classifier H_i.
- For each training mention:
  - Try all possible labels.
  - Based on label choice, predict remaining labels using H_i.
  - **Compute global cost for each choice.**

20 dead, **15** injured in a **USAirways Boeing 747** crash.

Gold:	Fat.	Fat.	Oper.	A.Type	
H_i:	Fat.	→ Fat.	→ ~~NIL~~	→ ~~NIL~~	Cost: 2
		→ ~~Inj.~~	→ Oper.	→ A. Type	Cost: 1
		→ etc…			

- Use computed costs to train classifier H_{i+1}.

Evaluation

- Task: Reconstruct knowledge base given just flight numbers.
- Metric: Multiclass Precision and Recall
  - Precision: # correct (non-NIL) guesses / total (non-NIL) guesses
  - Recall: # slots correctly filled / # slots possibly filled

	Precision	Recall	F-score
Maj. Class	0.026	0.237	0.047
Local Model	0.187	0.370	0.248
SMLI	0.185	0.386	0.250
CRF Model	0.159	**0.425**	0.232
Searn Model	**0.240**	0.370	**0.291**

Feature Ablation

	Precision	Recall	F-score
All features	0.240	0.370	0.291
- location in document	0.245	0.386	0.300
- syntactic dependencies	0.240	0.330	0.278
- sentence context	0.263	0.228	0.244
- local context	0.066	0.063	0.064

Feature Ablation

	Precision	Recall	F-score
All features	0.240	0.370	0.291
- location in document	0.245	0.386	0.300
- syntactic dependencies	0.240	0.330	0.278
- sentence context	0.263	0.228	0.244
- local context	0.066	0.063	0.064

Feature Ablation

	Precision	Recall	F-score
All features	0.240	0.370	0.291
- location in document	0.245	0.386	0.300
- syntactic dependencies	0.240	0.330	0.278
- sentence context	0.263	0.228	0.244
- local context	0.066	0.063	0.064

Feature Ablation

	Precision	Recall	F-score
All features	0.240	0.370	0.291
- location in document	0.245	0.386	0.300
- syntactic dependencies	0.240	0.330	0.278
- sentence context	0.263	0.228	0.244
- local context	0.066	0.063	0.064

Summary

- New plane crash dataset and evaluation task.
- Distant supervision framework for event extraction.
- Evaluate several models in this framework.

Thanks!