

# Biomedical Event Extraction without Training Data

Joint work with Paula Buttery, Diarmuid Ó Séaghdha and Ted  
Briscoe

Andreas Vlachos

`Andreas.Vlachos@cl.cam.ac.uk`

Computer Laboratory  
University of Cambridge

08 May 2009



# Goals

- Domain-independence
- Avoid using the training data (cheaper to apply)
- Performance is unlikely to reach high levels, but:
  - Better familiarization with the task
  - Identify what training data/adaptation is needed
  - Better features for machine learning algorithms



# In a nutshell

Key idea: events are denoted by verbs and they are syntactically connected to their arguments

- 1 Identify candidate triggers
- 2 Connect triggers with arguments using the parser output
- 3 Resolve anaphoric arguments
- 4 Produce complete events



# Trigger identification

- Assumptions used for trigger identification:
  - Triggers are expected to be verbs or nominalizations
  - Each trigger verb denotes one event class exclusively
- Built a dictionary of verbs associated with event classes:
  - 1 Lemmatization (RASP)
  - 2 Stemming (Porter)
  - 3 Kept trigger stem-event class pairs appearing at least 10 times in the training data
  - 4 Mapped stems to verbs
- No other use of the training data in our system
- Such a dictionary could be obtained from an expert, possibly assisted by semantic clustering



## Trigger identification (cont.)

- Using the dictionary we attempt to map each token to a verb (if possible)
- Successive relaxation matching: lemmas, stems, partial matching
- Partial matching allows mapping:
  - “co-transfect” to “transfect”
  - “gp40-induced” to “induced”



# Trigger identification limitations

- Can't identify multi-token triggers
  - ⇒ Flexible matching evaluation allows for partial triggers
  - ⇒ Single-token triggers should suffice for event extraction
- Some verbs denote multiple event classes in the training data
  - ⇒ If consistent, generate one trigger per event class (e.g. “overexpress”)
  - ⇒ If inconsistent, generate a trigger for the most frequent class (e.g. “express”)
- Trigger overgeneration:
  - ⇒ will be dealt with at argument extraction (not all triggers will form events)
- Can't identify common triggers that cannot be mapped to a verb (e.g. “mrna”)



# Trigger identification limitations

- Can't identify multi-token triggers
  - ⇒ Flexible matching evaluation allows for partial triggers
  - ⇒ Single-token triggers should suffice for event extraction
- Some verbs denote multiple event classes in the training data
  - ⇒ If consistent, generate one trigger per event class (e.g. “overexpress”)
  - ⇒ If inconsistent, generate a trigger for the most frequent class (e.g. “express”)
- Trigger overgeneration:
  - ⇒ will be dealt with at argument extraction (not all triggers will form events)
- Can't identify common triggers that cannot be mapped to a verb (e.g. “mrna”)



# Trigger identification limitations

- Can't identify multi-token triggers
  - ⇒ Flexible matching evaluation allows for partial triggers
  - ⇒ Single-token triggers should suffice for event extraction
- Some verbs denote multiple event classes in the training data
  - ⇒ If consistent, generate one trigger per event class (e.g. “overexpress”)
  - ⇒ If inconsistent, generate a trigger for the most frequent class (e.g. “express”)
- Trigger overgeneration:
  - ⇒ will be dealt with at argument extraction (not all triggers will form events)
- Can't identify common triggers that cannot be mapped to a verb (e.g. “mrna”)





# Trigger identification limitations

- Can't identify multi-token triggers
  - ⇒ Flexible matching evaluation allows for partial triggers
  - ⇒ Single-token triggers should suffice for event extraction
- Some verbs denote multiple event classes in the training data
  - ⇒ If consistent, generate one trigger per event class (e.g. “overexpress”)
  - ⇒ If inconsistent, generate a trigger for the most frequent class (e.g. “express”)
- Trigger overgeneration:
  - ⇒ will be dealt with at argument extraction (not all triggers will form events)
- Can't identify common triggers that cannot be mapped to a verb (e.g. “mrna”)



# Argument Extraction

- Goal: Associate candidate triggers with appropriate arguments
- Rely on the output of the domain-independent RASP toolkit
- Fixed the PoS tags of the protein tokens to proper nouns
- Weighted Grammatical Relations (GR) output extracted from the top-10 parses:
- Sample output for: “We further show that Nm interacts with all STATs except Stat2 .”

weight	GR type	head	dependent
0.946554	ncmod	STATs:9_ NN2	except:10_ II
0.556104	ncmod	STATs:9_ NN2	all:8_ DB2
0.443896	ncmod	STATs:9_ NN2	all:8_ DB
1.0	dobj	with:7_ IW	STATs:9_ NN2
0.954550	iobj	interact+s:6_ VVZ	with:7_ IW

## Argument Extraction (cont.)

We associate triggers with arguments using these GR paths:

- verb-trigger  $\xrightarrow{\text{subject}}$  arg
- noun-trigger  $\xrightarrow{\text{ind.obj.}}$  prep  $\xrightarrow{\text{d.obj.}}$  arg
- noun-trigger  $\xrightarrow{\text{modifier}}$  arg
- trigger  $\xrightarrow{\text{modifier}}$  prep  $\xrightarrow{\text{object}}$  arg
- trigger  $\xrightarrow{\text{passivesubj.}}$  arg



## Argument Extraction (cont.)

- Using the dev. data we lexicalized the rules on the linking prepositions:  
one set for *Binding*, one for *Regulation* and one for simple events.
- We allow triggers to be *Regulation* event args
- For *Regulation* Theme/Cause distinctions were implemented
- Triggers were not lexicalized, only constrained by PoS
- Only partial events (trigger-arg pairs) are produced at this stage
- Partial events involving anaphoric mentions are processed further



# Anaphora resolution

Why anaphora:

- Events can span over more than one sentences
- Sentences can be too complex to extract events using parsing output only:  
“phosphorylation of cellular proteins, notably phospholipase C gamma 1”
- Coreference between events:  
The expression of LAL-mRNA is induced. This induction is dependent on. . .

Heuristic rules favouring precision over recall were implemented based on the error analysis of the development data



# Event post-processing

- Simple events were formed from suitable trigger-arg pairs
- *Regulation* events were formed by identifying suitable Themes iteratively
- Suitable Themes were proteins or completed events
- Cyclic partial events between triggers occurred due to the use of top-10 parses:
  - ⇒ Resolved using GR weights
- Causes were attached (if possible) to *Regulation* events with a Theme
- Only single-arg *Binding* events were formed



## Full results

Event Class	Development		Test	
	recall	precision	recall	precision
Localization	45.28	92.31	25.86	90.00
Binding	12.50	24.41	12.68	31.88
Gene expression	52.25	80.79	45.57	75.81
Transcription	42.68	77.78	12.41	56.67
Protein catabolism	42.86	81.82	35.71	83.33
Phosphorylation	63.83	78.95	49.63	77.91
Neutral regulation	20.12	50.75	9.28	36.49
Positive regulation	16.86	48.83	11.39	38.49
Negative regulation	11.22	36.67	6.86	36.11
Total	26.55	58.09	21.12	56.90
Binding (decomposed)	26.92	66.14	18.84	54.35

# Comments

- Higher Precision at the expense of Recall (expected)
- Overall F-scores: 36.44 on dev, 30.8 on test (10th out of 24)
- Highlight: Precision on simple events 75-92% (6th overall)
- Weakness: *Regulation* F-score: 24.21 (dev), 15.79 (test) (still 10th overall)
- *Regulation* events account for 48% of all the events





## More comments...

- For three simple event classes, a single trigger verb was identified:
  - *Phosphorylation*: “phosphorylate”
  - *Protein\_catabolism*: “degrade”
  - *Transcription*: “transcribe”
- Performances were good (50-78 F-score), apart from *Transcription* on test: 20%
- Common triggers could not be mapped to a verb (e.g. “mrna”)
- Common trigger “express” (100 times in training) considered a *Gene\_expression* trigger



## More comments...

- For three simple event classes, a single trigger verb was identified:
  - *Phosphorylation*: “phosphorylate”
  - *Protein\_catabolism*: “degrade”
  - *Transcription*: “transcribe”
- Performances were good (50-78 F-score), apart from *Transcription* on test: 20%
- Common triggers could not be mapped to a verb (e.g. “mrna”)
- Common trigger “express” (100 times in training) considered a *Gene\_expression* trigger



## More comments...

- For three simple event classes, a single trigger verb was identified:
  - *Phosphorylation*: “phosphorylate”
  - *Protein\_catabolism*: “degrade”
  - *Transcription*: “transcribe”
- Performances were good (50-78 F-score), apart from *Transcription* on test: 20%
- Common triggers could not be mapped to a verb (e.g. “mrna”)
- Common trigger “express” (100 times in training) considered a *Gene\_expression* trigger



# Lessons learnt

- Dictionary-based trigger identification works well enough but does not take context into account
- Rule-based argument extraction worked well for simple events but:
  - Lexicalization at the trigger level is needed
  - Event classes with more than one arguments are not handled properly
  - Distinction between different argument types is problematic
- Anaphora can be useful, but needs good performance at the previous stages



# Questions?

