CMU LTI @ KBP 2015 Event Track

Zhengzhong Liu
Dheeru Dua
Jun Araki
Teruko Mitamura
Eduard Hovy

LTI Carnegie Mellon University

Language
Technologies
Institute

Event Nugget Detection

Nugget Detection

1. Three tasks:
    a. Detect the spans that corresponds to event mentions
    b. Detect the event nugget type
    c. Detect the Realis Status
2. New Challenge:
    a. Double tagging

LTI1

1. Discriminatively trained CRF.
    a. Test with averaged perceptron
2. Handle double tagging by combining the multiple types into a new label.
3. Each nugget is predicted independently.

Combining Event Types?

Total Possible Joint Type: 34

Justice_Execute ; Life_Die	30
Transaction_Transfer-Ownership ; Movement_Transport-Artifact	27
Life_Die ; Conflict_Attack	48
Transaction_Transfer-Ownership ; Transaction_Transfer-Money	21
Conflict_Attack ; Life_Die	69
Justice_Extradite ; Movement_Transport-Person	39

Combining Event Types?

- You can even infer the text by looking at the types.
  - Smuggling (all 3 types all the time)
    - Transaction_Transfer-Money ; Movement_Transport-Artifact ; Transaction_Transfer-Ownership
  - Conflict_Attack ; Transaction_Transfer-Ownership
    - Hijacking, rob, burglary, seize
- Nugget Type detection is similar to WSD with the detailed ontology.
- Joint type should share information with its original types.
  - So the features are extracted on both the joint and splitted version

LTI 1 Features

- Standard Linguistic Features:
  - Part-of-Speech, lemma, named entity tag of the following:
    - The 2-word window of the trigger (both side)
    - The trigger word itself
    - Direct dependent words of the trigger
    - Dependent head of the trigger
- Ontology:
  - Brown clusters (8, 12, 16 bits)
  - WordNet Synonym and Noun derivative forms of the trigger
  - FrameNet Type

See our system at the end for details

LTI1 Features

- Selected WordNet senses in the context:
  - "Leader", "Worker", "Body Part", "Monetary System", "Possession", "Government", "Crime" and "Pathological State" (More on this later)
  - Whether surrounding words match such sense
  - Whether argument of mention match such sense (arguments from semantic roles)
- Semantic role features:
  - The frame name (mentioned above)
  - The argument's role, named entity tag, and headword lemma

See our system at the end for details

LTI2

1. CRF trained with Passive-Aggressive Perceptron.
2. Multi-tagging handling:
    a. Merging sequence from the top 5 series
    b. Training: Optimize top 5-best sequence

LTI2

1. Normalized the top scores and take the largest gap.
2. p=0.4, ε is 0.01.

Input: 5-Best-Viterbi-Parses $(y_j, score_j), 1 \leq j \leq 5$
Input: Threshold, ϵ and Scaling Factor p
Output: Top-n Predicted Sequence

$prediction \leftarrow \{y_1\}$
$\mu \leftarrow \frac{\sum_{i=1}^{5} score_i}{5}$
$\sigma \leftarrow \sqrt{\frac{\sum_{i=1}^{5} (score_i - \mu)^2}{5}}$
for $j \leftarrow 2...5$ **do**
  $Nscore_j = [(score_j - \mu)/\sigma] * p$
  if $Nscore_j - Nscore_{j+1} \leq \epsilon$ **then**
    $prediction \leftarrow prediction \cup y_j$
  else
    return $prediction$

LTI2: Features

- POS tags,in the 5 word window.
- Ontology:
  - Brown clusters with 13 bits
  - Lemmas of the event trigger in the WordNet hierarchy
- History:
  - 2 verbs in past and future
  - 2 events trigger seen in the history
- Event arguments types from SRL followed by NER of the arguments.
- Recall Mode:
  - 8 bit Brown cluster, a gazetteer of event triggers and WordNet synsets

Realis Classification

1. Linear SVM model.
2. Basic features are borrowed from type detection:
   a. All lexicalized features are removed to avoid overfitting
   b. One feature to see if the phrase is **“in quote”**
3. Done after span and type detection.

See our system at the end for details

Results (LTI1 on Dev, 5-fold aver.)

	Precision	Recall	F1
Plain	74.36	55.722	63.622
Type	67.08	50.25	57.382
Realis	51.788	38.754	44.274
Type+Realis	46.288	34.626	39.562

Results (Realis On Dev with Gold Mentions)

- Realis itself is difficult.
- It is more serious with imperfect mention types.

	Prec	Recall	F1
Fold 1	71.68	71.63	71.66
Fold 2	64.06	64.06	64.06
Fold 3	62.07	61.96	62.02
Fold 4	72.66	72.66	72.66
Fold 5	62.21	62.21	62.21
Aver.	**66.536**	**66.504**	**66.522**

Final Results on Evaluation Set

LTI1	Prec	Recall	F1
Plain	82.46	50.3	62.49
Type	73.68	44.94	55.83
Realis	62.09	37.87	47.05
All	55.12	33.62	41.77

LTI2	Prec	Recall	F1
Plain	77	39.53	52.24
Type	68.79	35.31	46.67
Realis	51.41	26.39	34.88
All	45.47	23.34	30.85

Results after fixing LTI2 format error

LTI2-Prec	Prec	Recall	F1
Plain	81.7	44.36	57.52
Type	72.91	39.56	51.29
Realis	61.84	33.55	43.50
All	55.37	30.04	38.9

LTI2-Recall	Prec	Recall	F1
Plain	77.59	49.14	60.17
Type	69.61	44.08	53.98
Realis	52.71	38.38	40.87
All	47.17	29.87	36.58

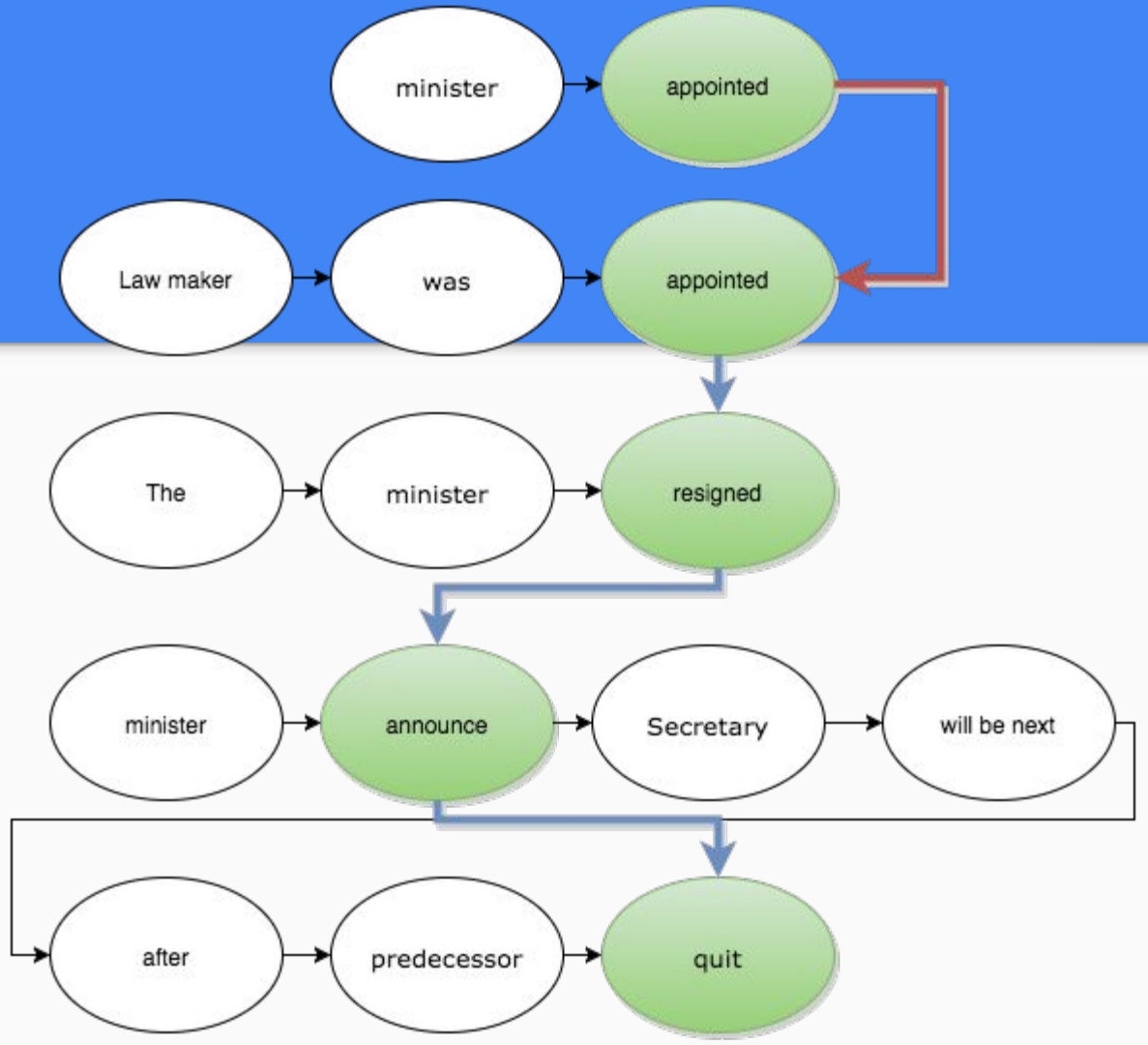
Future work

1. Hand selected WordNet senses can be replaced by statistical methods
   a. NPMI between WordNet Sense and the type:

census	Life_Divorce	0.6645
harassment	Justice_Sue	0.6641
declaration	Justice_Charge-Indict	0.6636
manufacturer	Manufacture_Artifact	0.6611
destination	Life_Marry	0.6595
government	Justice_Appeal	0.2502

Future work

1. Model inter-mention dependencies.
2. And of course, continuous representation can be helpful.



Event Hopper Coreference

Hopper Coreference

1. Identify Full Event Coreference links.
2. Given Information :
    a. Event Nuggets given, including the span, Event types and subtypes, and Realis
3. 2 Individual system with 3 submissions.
    a. We focus on our best system in the presentation

The Model

1. Latent Antecedent Tree
2. Represent cluster as a tree.
   a. Note that a coreference can be represented as multiple trees
3. Best First Decoding
   a. Favor “easy” decisions
   b. Ng & Cardie 2002

Input: Training data D, number of iterations T
Output: Weight vector w

$w = \vec{0}$
for $t \leftarrow 1..T$ **do**
  for $\langle M_i, \mathcal{A}_i, \tilde{\mathcal{A}}_i \rangle \in D$ **do**
    $\hat{y}_i = \arg\max_{\mathcal{Y}(\mathcal{A})} score(y)$
    if $\neg Correct(\hat{y}_i)$ **then**
      $\tilde{y}_i = \arg\max_{\mathcal{Y}(\tilde{\mathcal{A}})} score(y)$
      $\Delta = \Phi(\tilde{y}_i) - \Phi(\hat{y}_i)$
      $\tau = \frac{\Delta * w}{||\Delta||^2}$
      $w = w + \tau\Delta$
return w

Fernandes et. al. 2012; Björkelund & Kuhn 2014

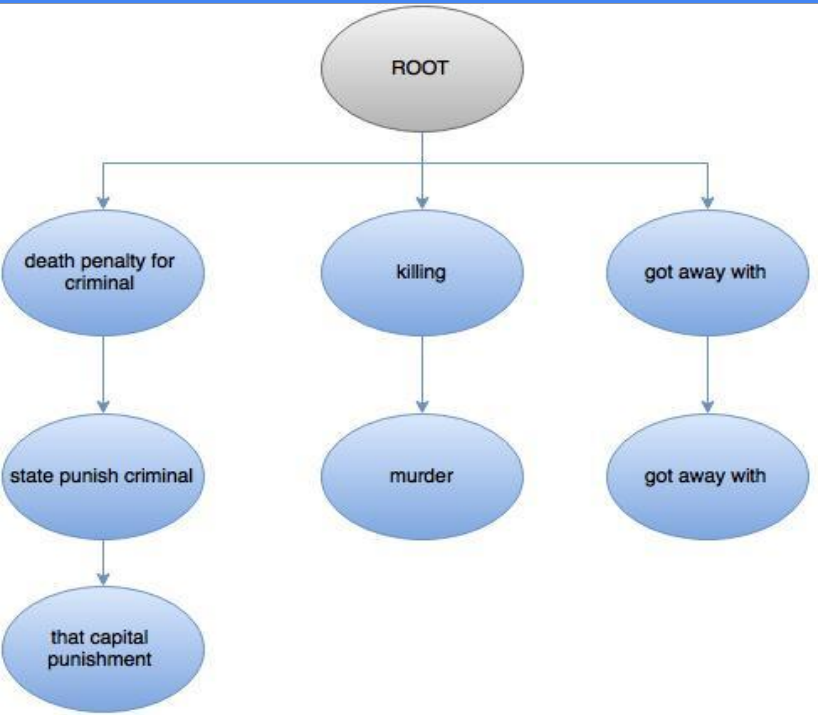
The LAT model

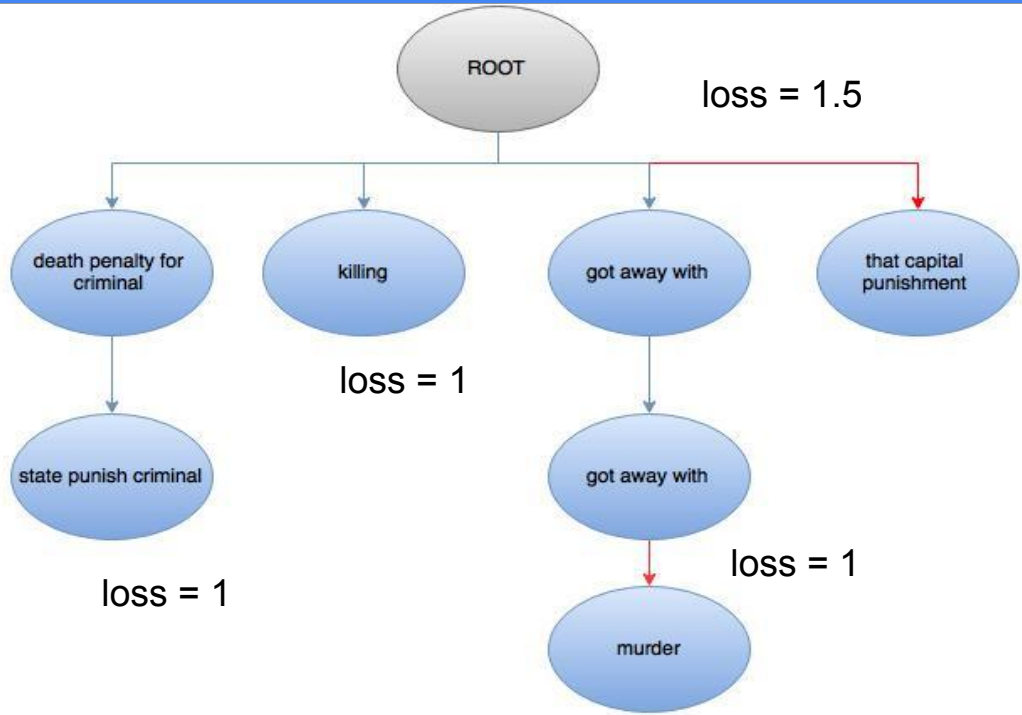
1. The Gold Tree:
   a. The best tree under current parameters
2. Predicted Tree:
   a. Prediction made with the Best-First algorithm
3. If clusters are difference, then penalize.
4. Trained with Passive Aggressive (Crammer et al. 2006).

Input: Training data D, number of iterations T
Output: Weight vector w

$w = \vec{0}$
for $t \leftarrow 1..T$ **do**
  for $\langle M_i, \mathcal{A}_i, \tilde{\mathcal{A}}_i \rangle \in D$ **do**
    $\hat{y}_i = \arg\max_{\mathcal{Y}(\mathcal{A})} score(y)$
    if $\neg Correct(\hat{y}_i)$ **then**
      $\tilde{y}_i = \arg\max_{\mathcal{Y}(\tilde{\mathcal{A}})} score(y)$
      $\Delta = \Phi(\tilde{y}_i) - \Phi(\hat{y}_i)$
      $\tau = \frac{\Delta * w}{||\Delta||^2}$
      $w = w + \tau\Delta$
return w

The LAT model





Features for coreference

1. Trigger Match - exact and fuzzy match on the trigger word
    a. uses standard linguistic features (pos, lemma, etc.)
    b. resources like Brown Clustering and WordNet.
    c. Information from mention type and realis type are also used
2. Argument match - exact and fuzzy match on the arguments
    a. String matches (head word, substring)
    b. Argument role
    c. Entity coreference information (From stanford)
3. Discourse features
    a. encodes sentence and mention distances

See our system at the end for details

Catch 1: The Importance of PA-algorithm

1. Passive Aggressive algorithm capture the loss term
   a. Penalize more if the tree differs a lot
2. We found that without using the PA-algorithm, it is hard to converge
3. Observations:
   a. Most clusters predictions are wrong -> Update is done almost all the time
   b. Some features differs between Forum dataset and News dataset -> e.g. Distance between mentions

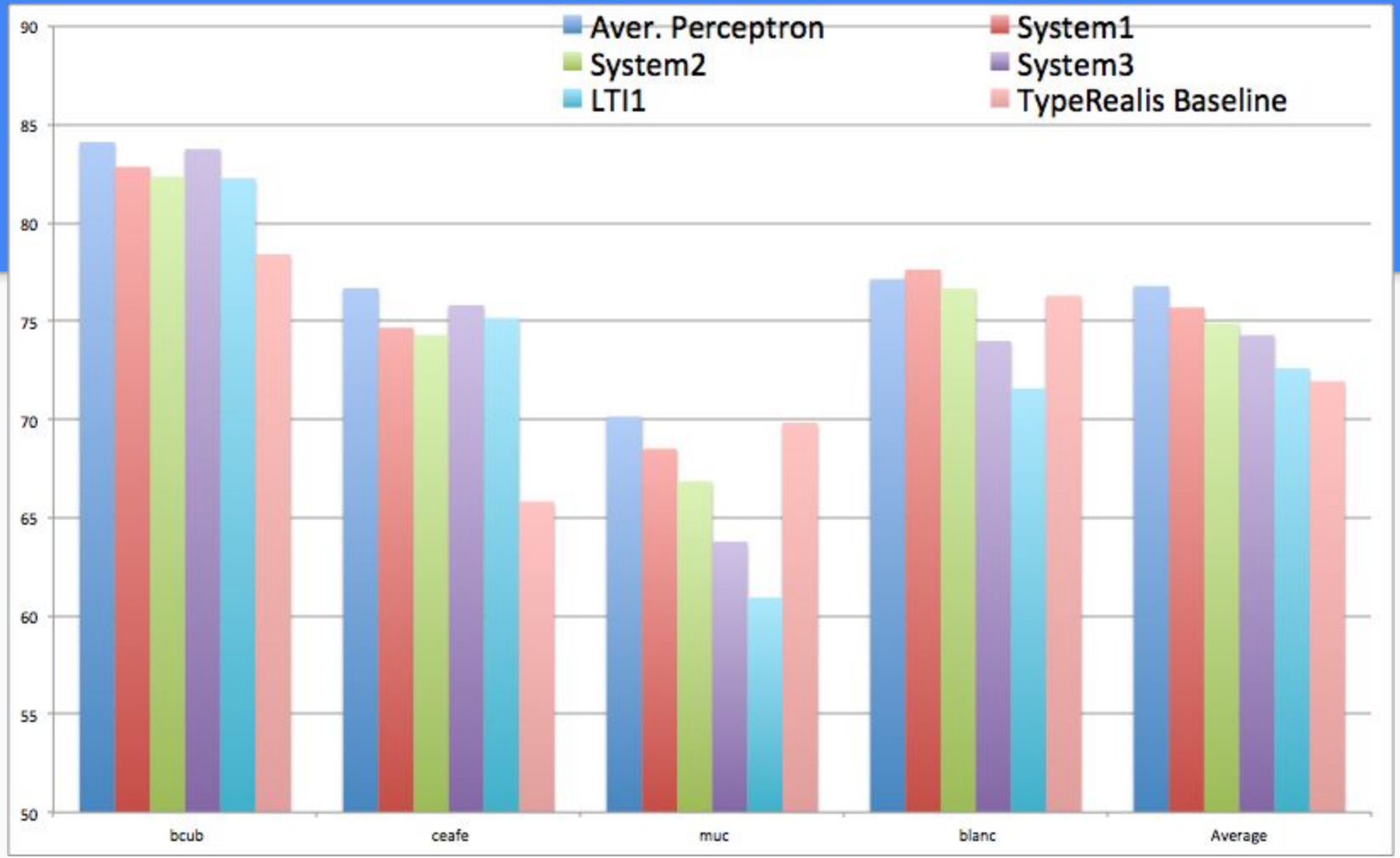
Catch 2: Averaging parameters matters

1. During training, we found different training sequence change the final model a lot.
2. However, the change is small with averaged perceptron.
3. Averaged score is also much better.

- Both problems might be caused by the data (i.e. multi-genre data without considering their differences)

5-fold results (Averaged vs. Vanilla)

	Average Perceptron	Vanilla Perceptron
CV0	83.08	79.16
CV1	78.53	72.72
CV2	75.80	75.13
CV3	77.15	69.63
CV4	74.20	61.94
Average	**77.75**	71.71


Aver. Perceptron
System1
System2
System3
LTI1
TypeRealis Baseline
90
85
80
75
70
65
60
55
50
bcub
ceafe
muc
blanc
Average

Off-cycle Evaluation (Full Pipeline)

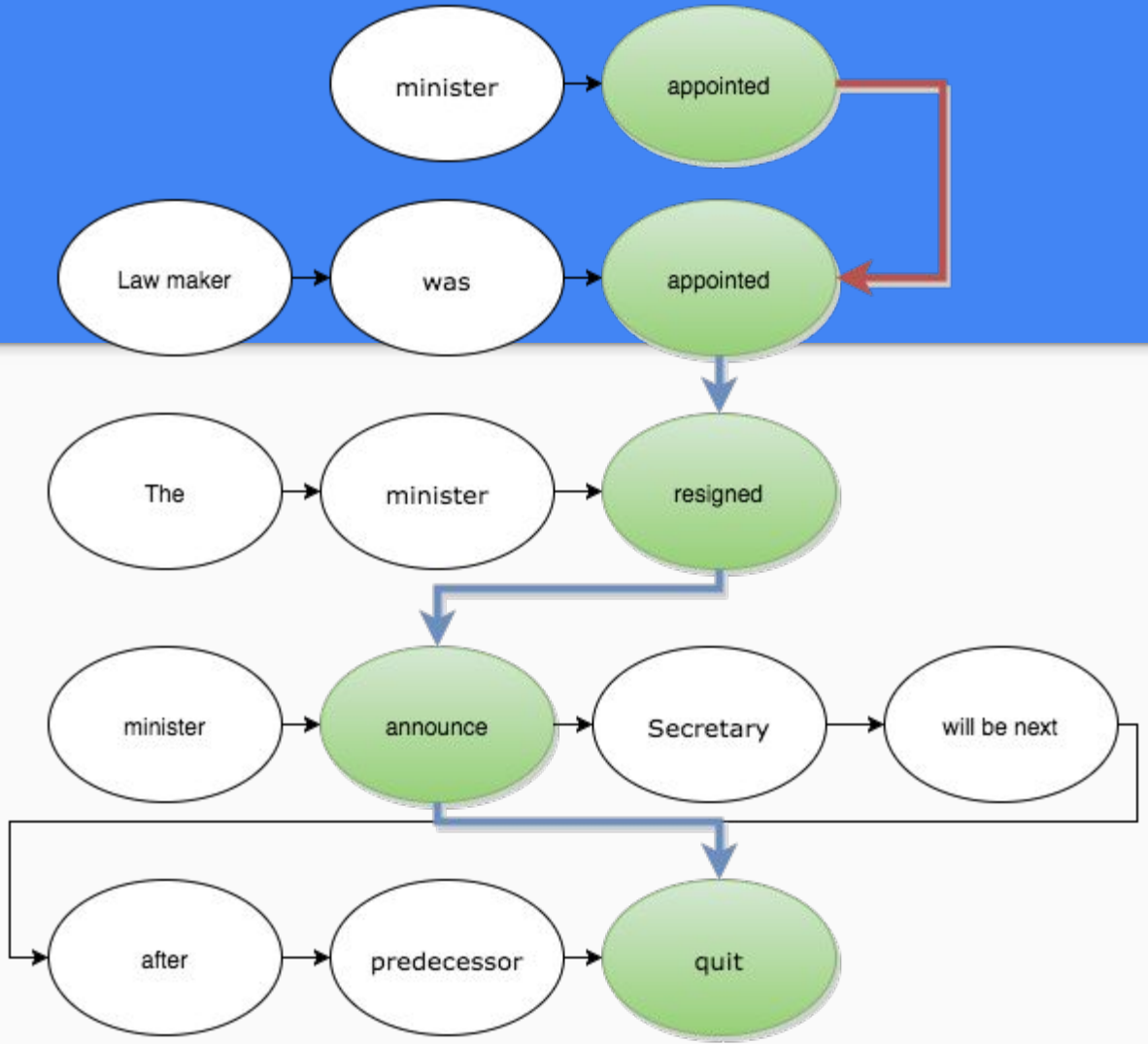
	BCubed	Ceafe	MUC	BLANC	Average
OUR_PIPELINE	**73.01**	**65.41**	**59.10**	**59.33**	**64.72**
System 1	69.65	64.55	56.86	59.51	63.23
System 2	67.27	61.35	63.93	58.52	62.95
System 3	68.28	61.99	61.85	58.05	62.80
System 4	67.80	61.62	62.30	57.79	62.63

Future Work

1. Consider genre specific features.
    a. We might train each genre independently
    b. Even better, consider only those features that might be affected by the genres (see next slide)
    c. For example, you will find a mention per 13.6 tokens in news but 25.3 tokens in forum.
2. Consider global features.
    a. It is not yet clear what global features can be useful to hopper coreference

Future Work

1. Consider interactions between mention detection.
2. Consider discourse level analysis.



Thank You!
Questions?

Our code here!

Might be hard to set up, but you can still have a look!

We are also working to integrate it into the DEFT project.

References

Anders Björkelund and Jonas Kuhn. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 47–57.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. *Joint Conference on {EMNLP} and {CoNLL-Shared} Task*:41–48.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, number July, pages 104–111, Philadelphia.