Identifying Constant and Unique Relations by using Time-Series Text

Yohei Takaku **Nobuhiro Kaji** **Naoki Yoshinaga** **Masashi Toyoda**

Toyo Keizai Inc. *Institute of Industrial Science, University of Tokyo*

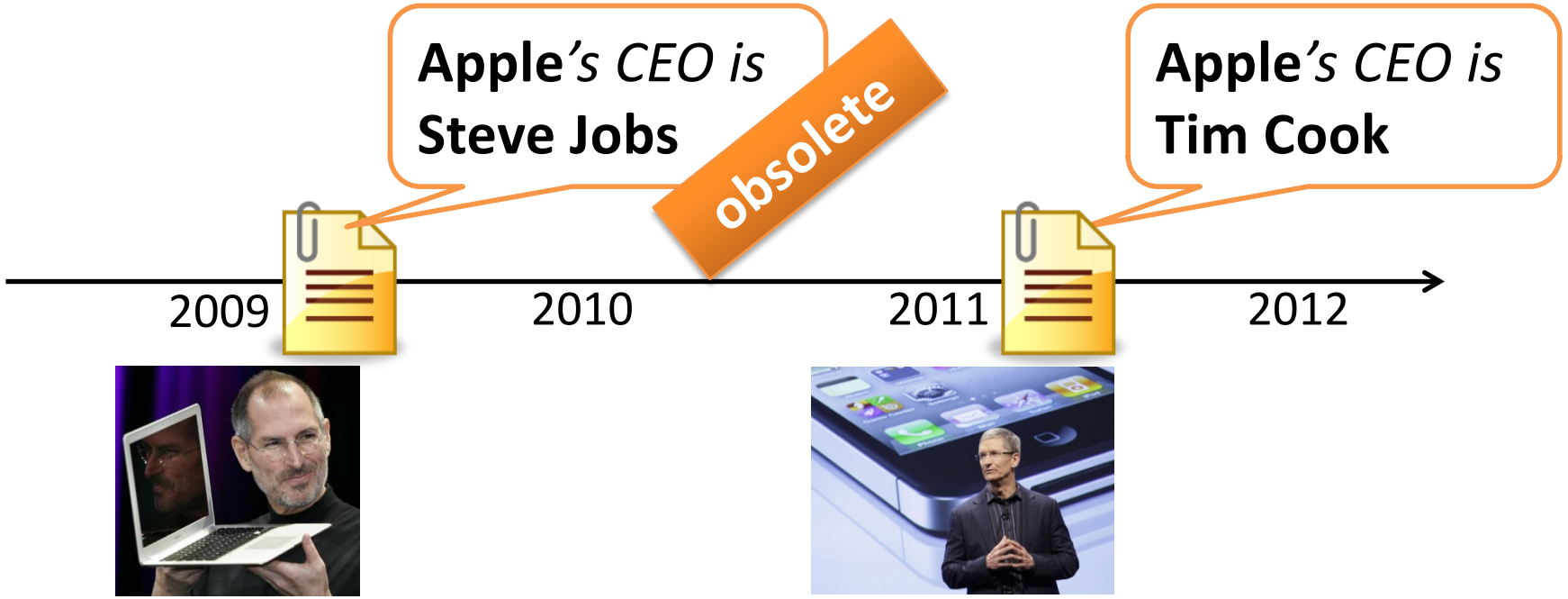
Relation Extraction from the Evolving Web

- **Web (text) as a growing goldmine** for extracting relations between real-world entities

  [Pantel+ 06; Banko+ 07; Suchanek+ 07; Wu+ 08,10; Zhu+ 09; Mintz+ 09]

  – Processing more text leads to more relations, **but**

  – **Relations in text could be obsolete / will become outdated**



How to Consistently Compile Extracted Relations?

- <**arg1**, *flows through*, **arg2**>
- <**arg1**, ‘s CEO is, **arg2**>
- <**arg1**, *sells*, **arg2**>

How to Consistently Compile Extracted Relations?

- <**arg1**, *flows through*, **arg2**>
- <**arg1**, ‘s CEO is, **arg2**>
- <**arg1**, *sells*, **arg2**>

Accumulate all the relations, **because the relation does not evolve over time**

How to Consistently Compile Extracted Relations?

- <**arg1**, *flows through*, **arg2**>
- **<arg1**, 's CEO is, **arg2>**
- **<arg1**, *sells*, **arg2>**

Overwrite with new one, **because the relation can take one value of arg2**

Constant and Unique Relations

- Given value of arg1**,**
  - **Constant rel.**: value of arg2 is **independent of time**
  - **Unique rel.**: value of arg2 is **one at any point in time**

constant, unique	**non-constant, unique**
<**arg1**, *was born in*, **arg2**>	<**arg1**, *'s CEO is*, **arg2**>
<**arg1**, *'s father of*, **arg2**>	<**arg1**, *belongs to*, **arg2**>
<**arg1**, *flows through*, **arg2**>	<**arg1**, *sells,* **arg2**>
constant, non-unique	**non-constant, non-unique**

Overview

- Constancy and Uniqueness of Relations
- Our Approach
- Features for Constancy Classification
- Features for Uniqueness Classification
- Experiments
- Conclusion

Two Binary Classification Tasks

<**arg1**, *was born in*, **arg2**>

<**arg1**, *'s father is*, **arg2**>

<**arg1**, *borders on*, **arg2**>

<**arg1**, *'s CEO is*, **arg2**>

<**arg1**, *belongs to*, **arg2**>

<**arg1**, *sells,* **arg2**>

Two Binary Classification Tasks

Task 1: constancy classification

constant	non-constant
<**arg1**, *was born in*, **arg2**>	<**arg1**, *'s CEO is*, **arg2**>
<**arg1**, *'s father is*, **arg2**>	<**arg1**, *belongs to*, **arg2**>
<**arg1**, *borders on*, **arg2**>	<**arg1**, *sells*, **arg2**>

Two Binary Classification Tasks

Task 2: uniquness classification

unique

<**arg1**, *was born in*, **arg2**>

<**arg1**, *'s CEO is*, **arg2**>

<**arg1**, *'s father is*, **arg2**>

<**arg1**, *belongs to*, **arg2**>

<**arg1**, *borders on*, **arg2**>

<**arg1**, *sells,* **arg2**>

non-unique

Two Kinds of Features for Training Supervised Classifiers

- Frequency obtained from time-series text
  - Detailed later
  - Based on blog posts crawled from 2006 to 2011

- Linguistic cues

  e.g., <**arg1**, *'s president is*, **arg2**>$_{\text{non-const.}}$

  Prefix George Bush is ex-president of USA

  e.g., <**arg1**, *borders on*, **arg2**>$_{\text{non-uniq.}}$

  Coordination France borders on Italy as well as Spain

Overview

- Constancy and Uniqueness of Relations
- Our Approach
- Features for Constancy Classification
- Features for Uniqueness Classification
- Experiments
- Conclusion

Using Time-series Text for Constancy Classification

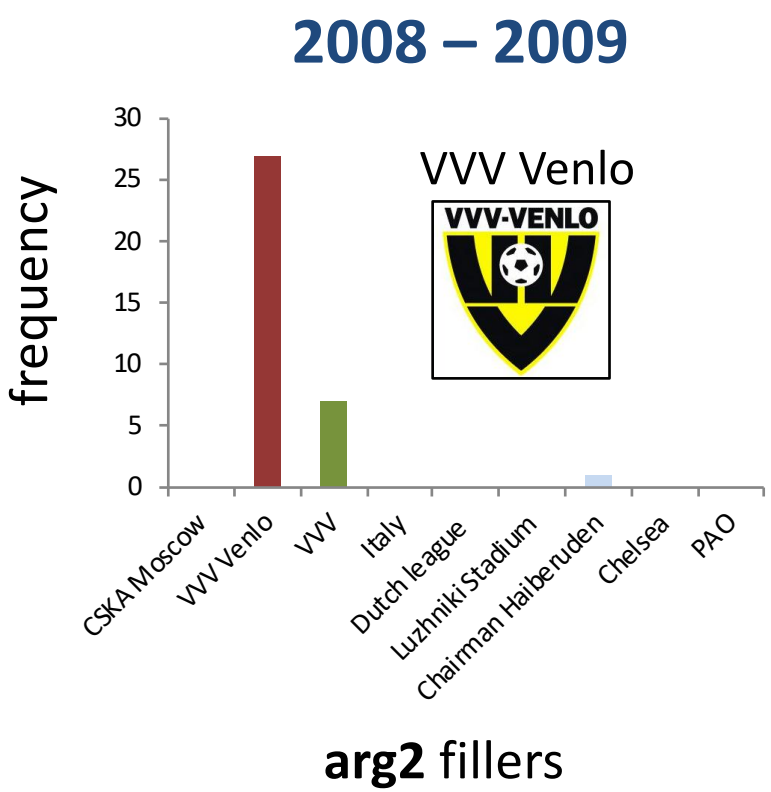
<Keisuke Honda(=arg1), *belongs to*, **arg2>**$_{\text{non-const.}}$

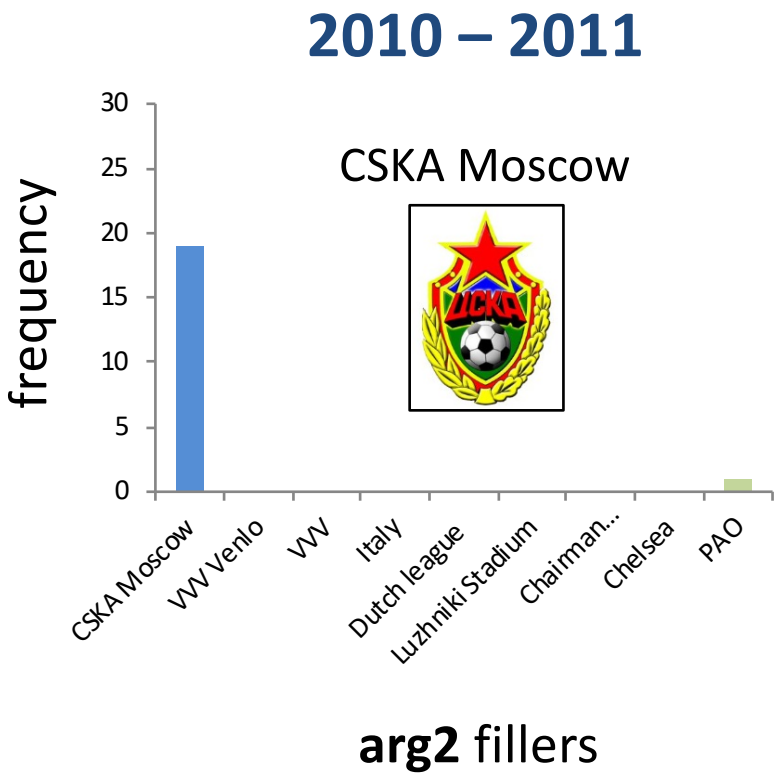
2008 – 2010, VVV-Venlo

2010 – now, CSK Moskow

Using Time-series Text for Constancy Classification

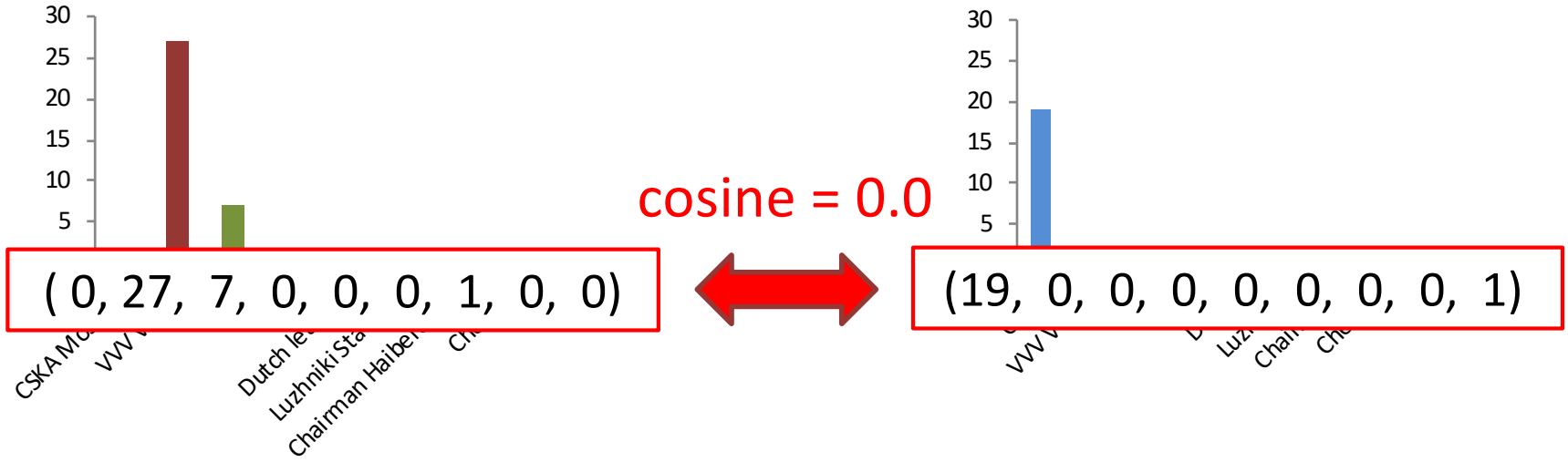
<**Keisuke Honda(=arg1)**, *belongs to*, **arg2**>non-const.



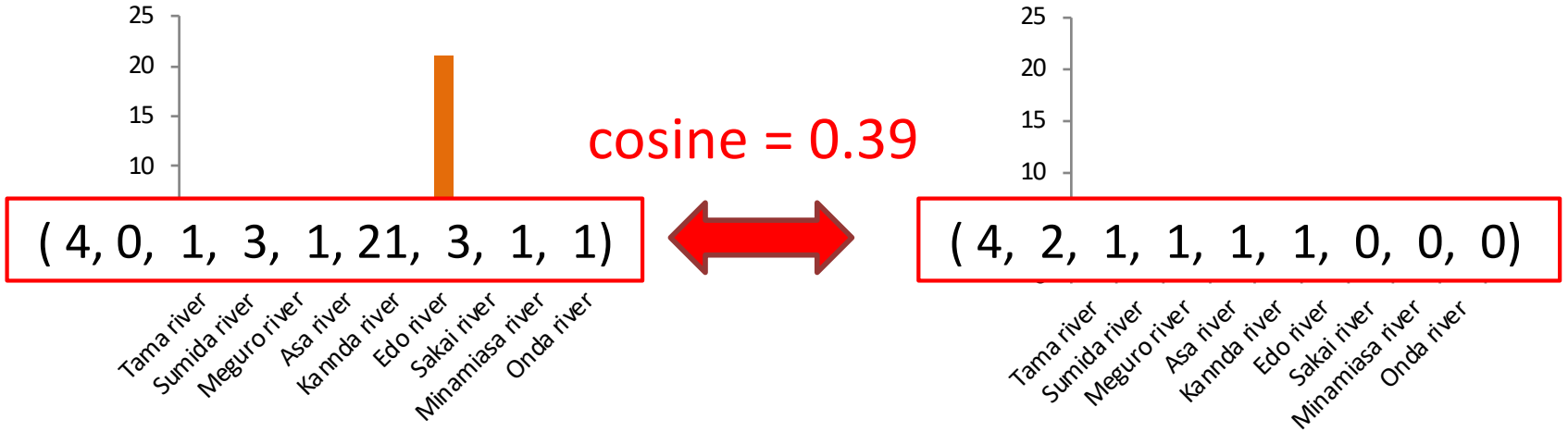


Cosine Similarity as a Feature Value

<Keisuke Honda, *belongs to*, arg2>non-const.

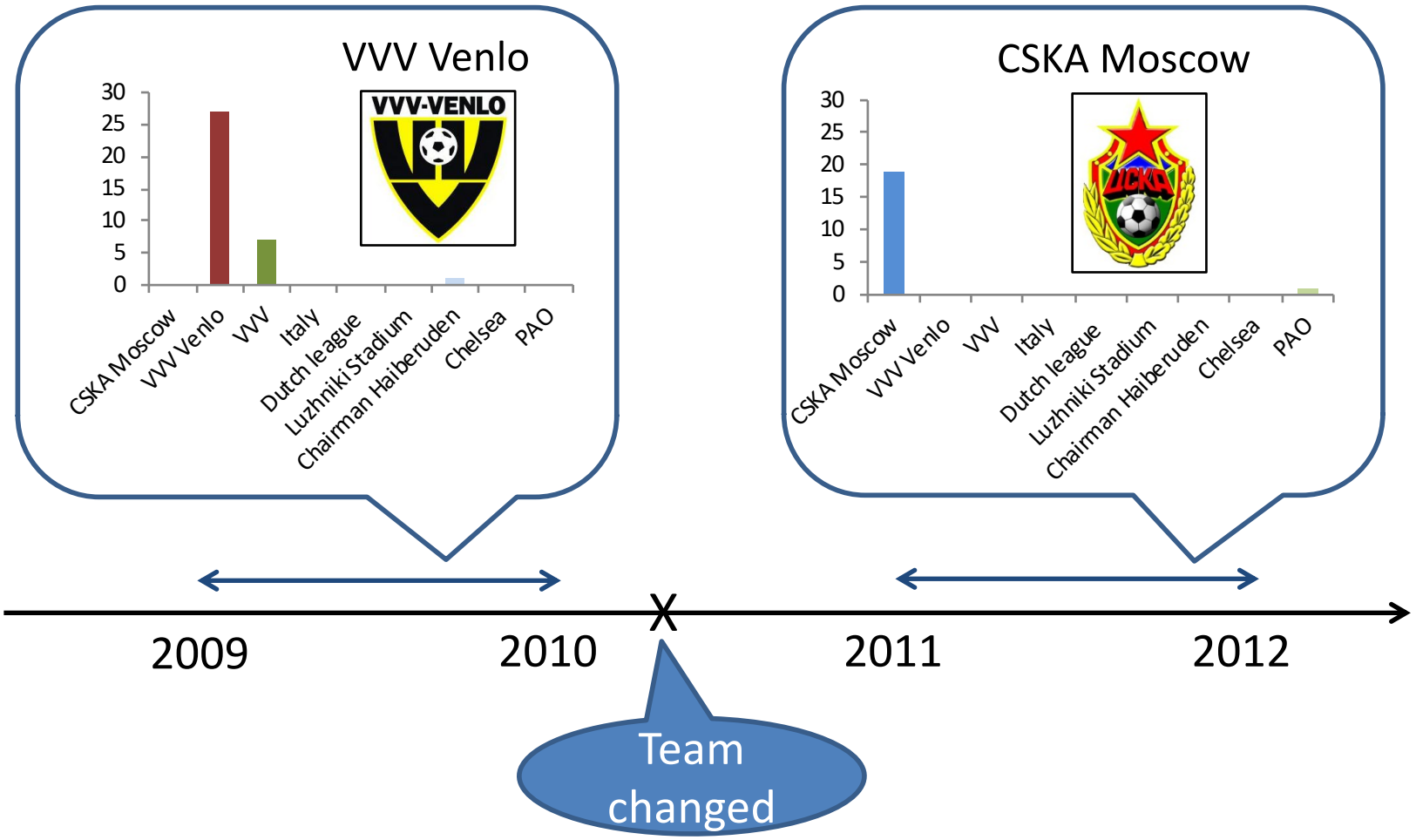


<Tokyo, *has river*, arg2>const.



Importance of Choosing Time Windows

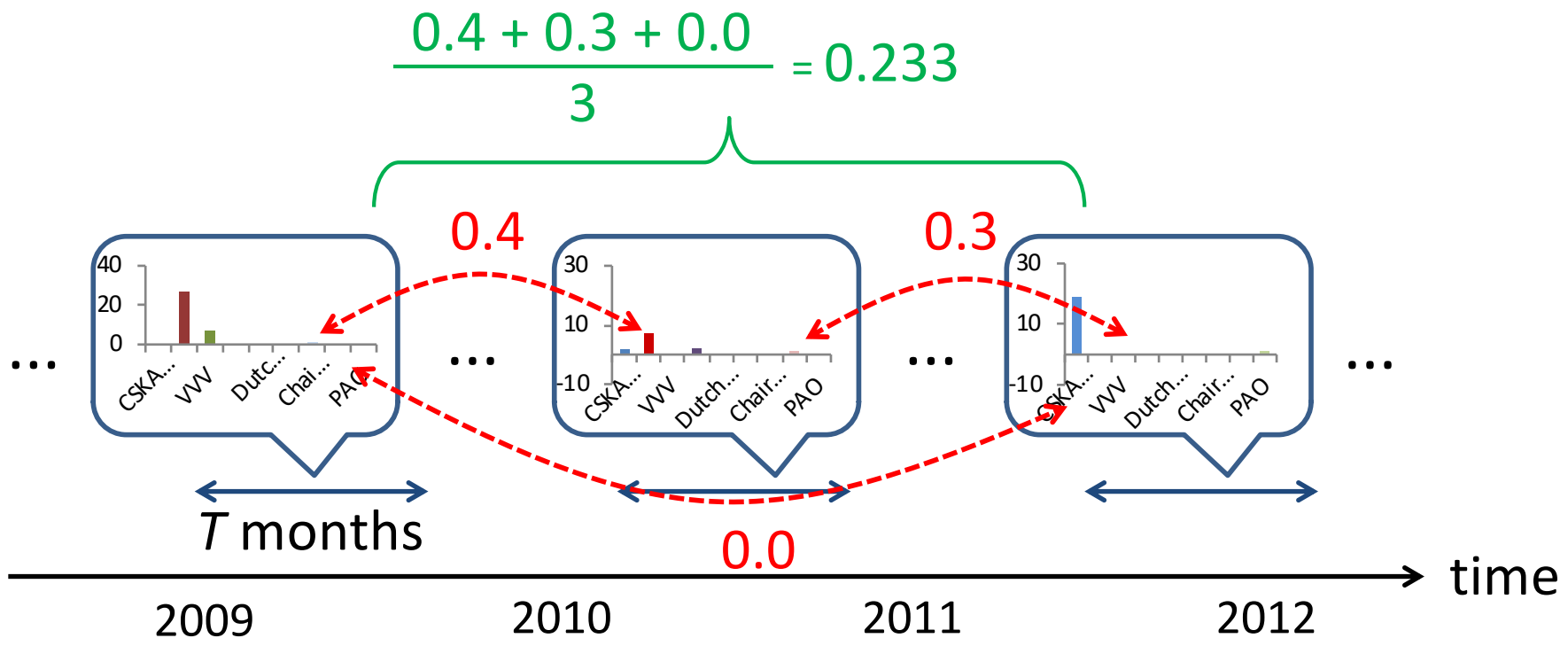
<**Keisuke Honda**, *belongs to*, **arg2**>non-const.



Importance of Choosing Time Windows

<**Keisuke Honda**, *belongs to*, **arg2**>non-const.

Using Multiple Time Windows



Window size T = { 1, 3, 6, 12 (months) }

Integration method = { ave., min., max. }

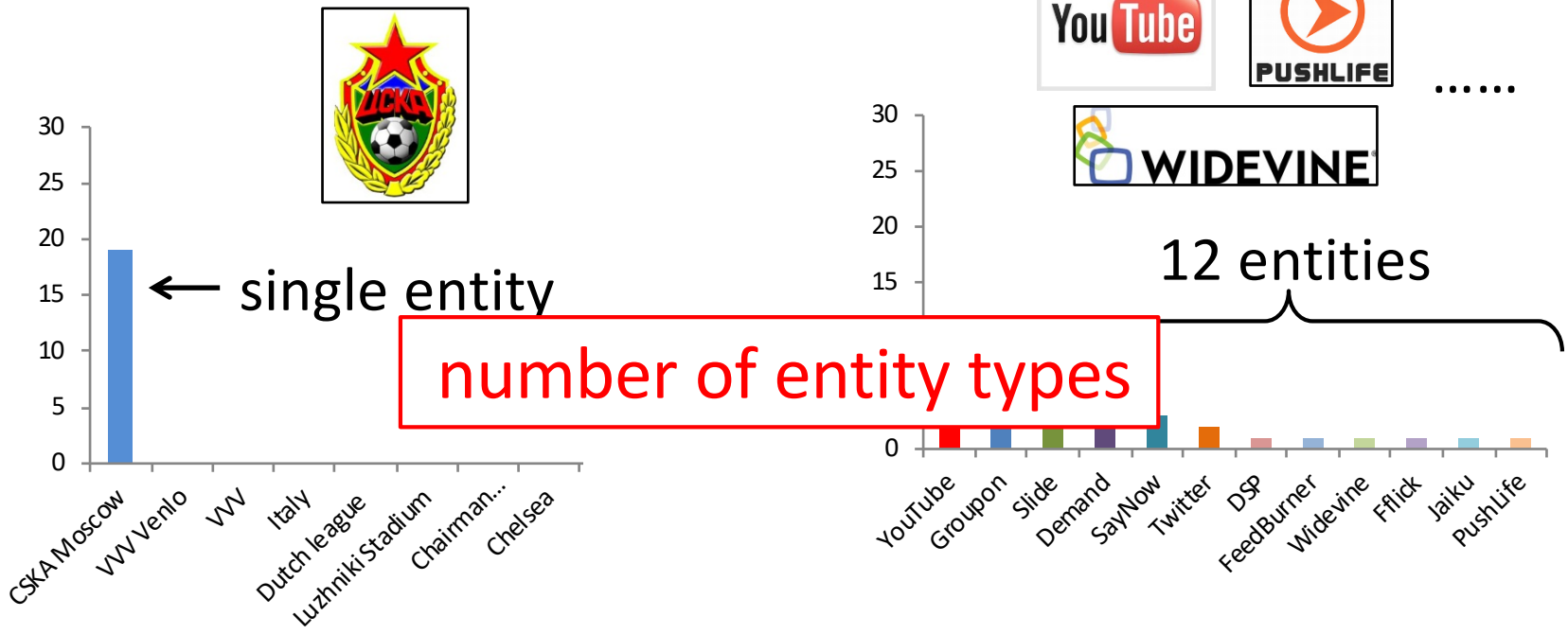
⇨ 12 (= 4 x 3) features

Overview

- Constancy and Uniqueness of Relations
- Our Approach
- Features for Constancy Classification
- Features for Uniqueness Classification
- Experiments
- Conclusion

Using Time-series Text for Uniqueness Classification

<**Keisuke Honda**, *belongs to*, **arg2**>$_{\text{uniq.}}$ <**Google**, acquires, **arg2**>$_{\text{non-uniq.}}$



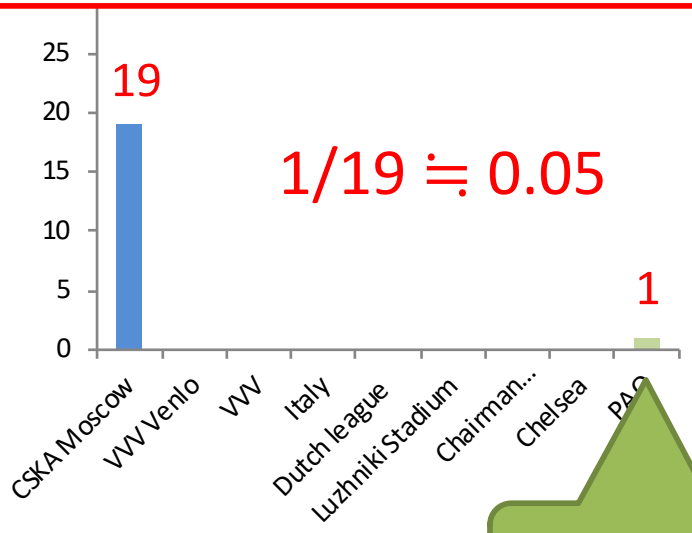
arg2 fillers

arg2 fillers

Using Time-series Text for Uniqueness Classification (Cont.)

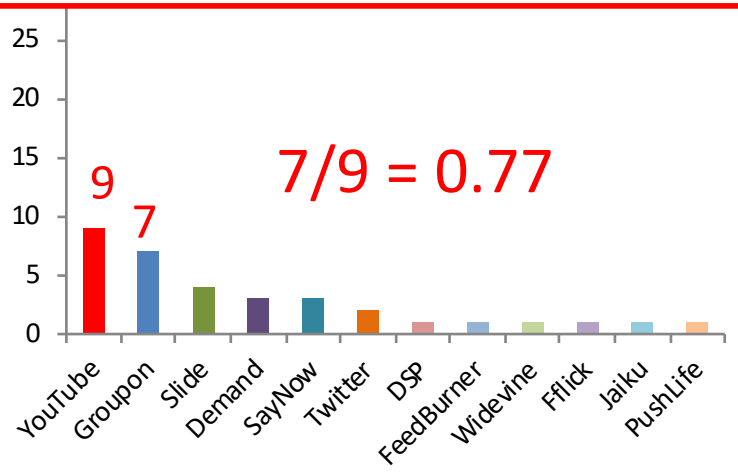
<**Keisuke Honda**, *belongs to*, **arg2**>$_{\text{uniq.}}$ <**Google**, acquires, **arg2**>$_{\text{non-uniq.}}$

Frequency ratio between the 1st and 2nd most frequent entities



values of **arg**

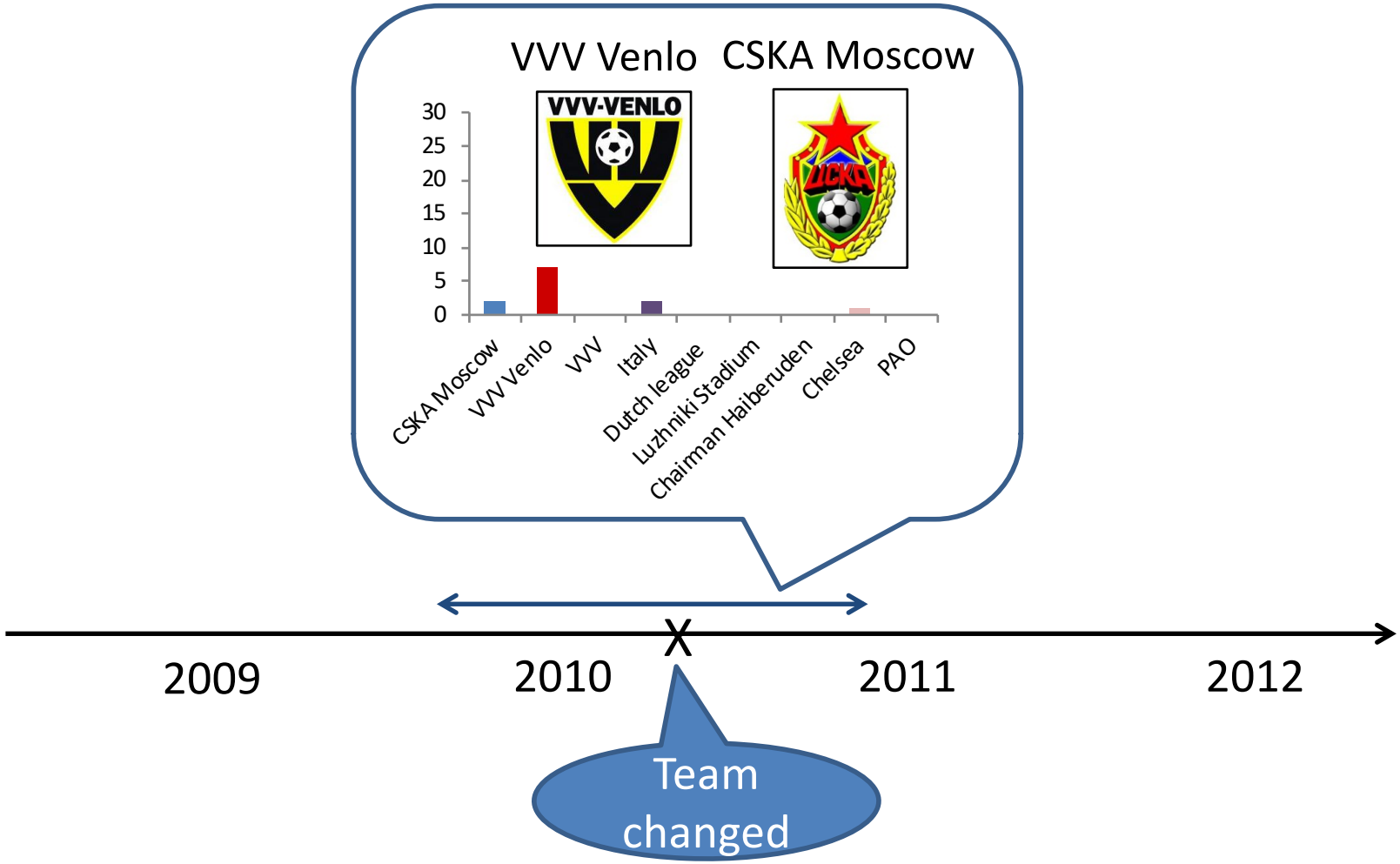
data can be noisy



values of **arg2**

Setting Appropriate Time Windows is also Important

<**Keisuke Honda**, *belongs to*, **arg2**>$_{\text{uniq.}}$



Overview

- Constancy and Uniqueness of Relations
- Our Approach
- Features for Constancy Classification
- Features for Uniqueness Classification
- Experiments
- Conclusion

Experiments

- Evaluate our method with relations extracted from time-series text
  - **Classifier:** Passive-aggressive algorithm w/ proposed features
  - **Time-series text:** 6-year's worth of Japanese blog posts (2.3-billion sentences)

- Conduct experiments to:
  - Evaluate the constancy classification
  - Evaluate the uniqueness classification
  - Investigate the impact of multiple window sizes

Data and Settings

- Parse time-series text to extract dependency paths connecting two named entities as relation instances

- Annotate 1000 relations (majority vote, 3 humans)

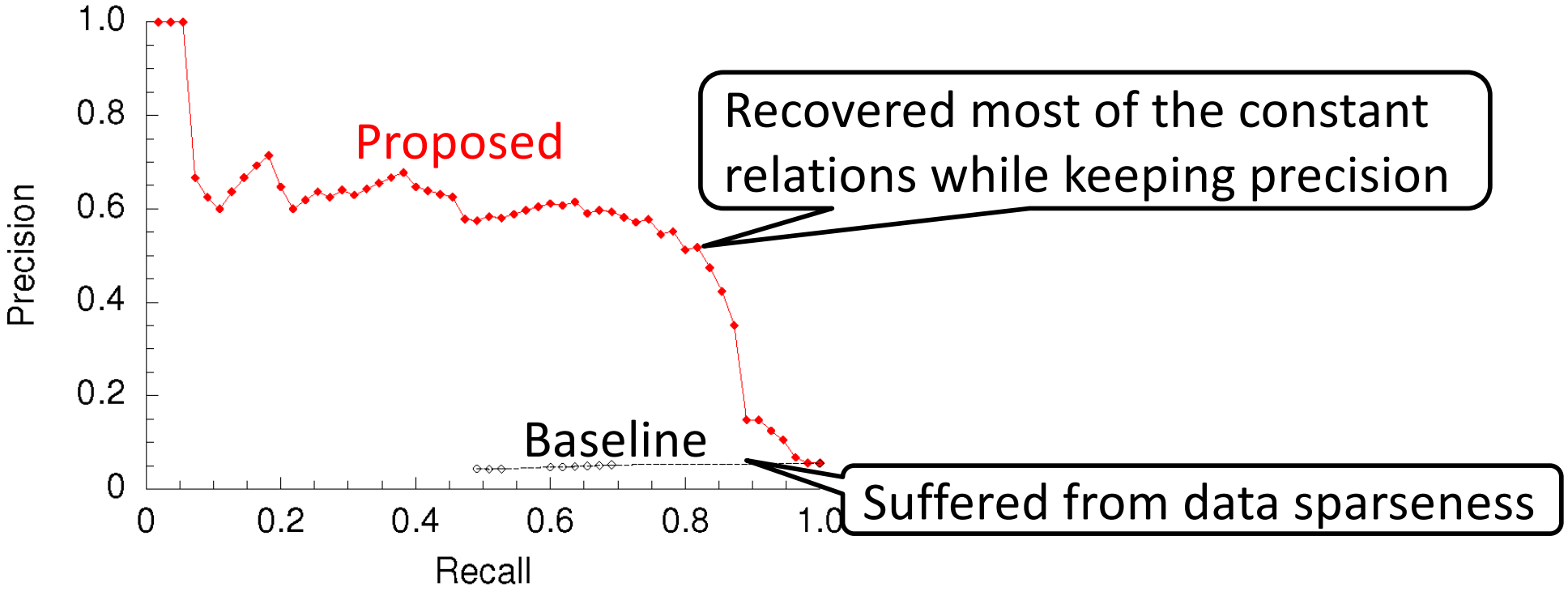
	Constancy	Uniqueness
Kappa [Fleiss 1971]	0.346 (fair)	0.428 (moderate)

  - Major reason for disagreement: *type* ambiguity
    Ex. <**arg1**$_{human}$, *is seen in*, **arg2**>$_{\text{non-const.}}$
    <**arg1**$_{mountain}$, *is seen in*, **arg2**>$_{\text{const.}}$

- Use the labeled relations for training & testing
  - Evaluation metric: Precision & Recall (5-fold cross validation)

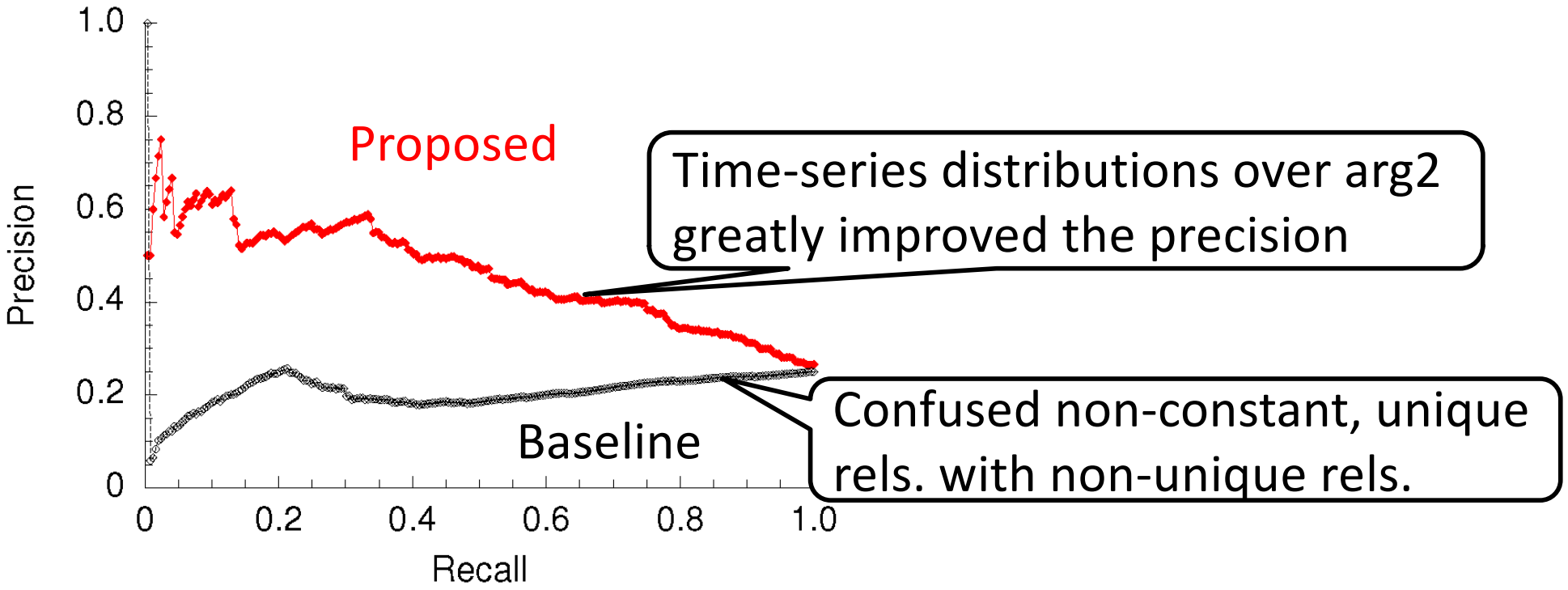
Classification Result: Constancy

- Varying the threshold to classifier's output (margin) to plot recall-precision curve
  - **Baseline:** cosine similarity between distributions over arg2 in the first and last month



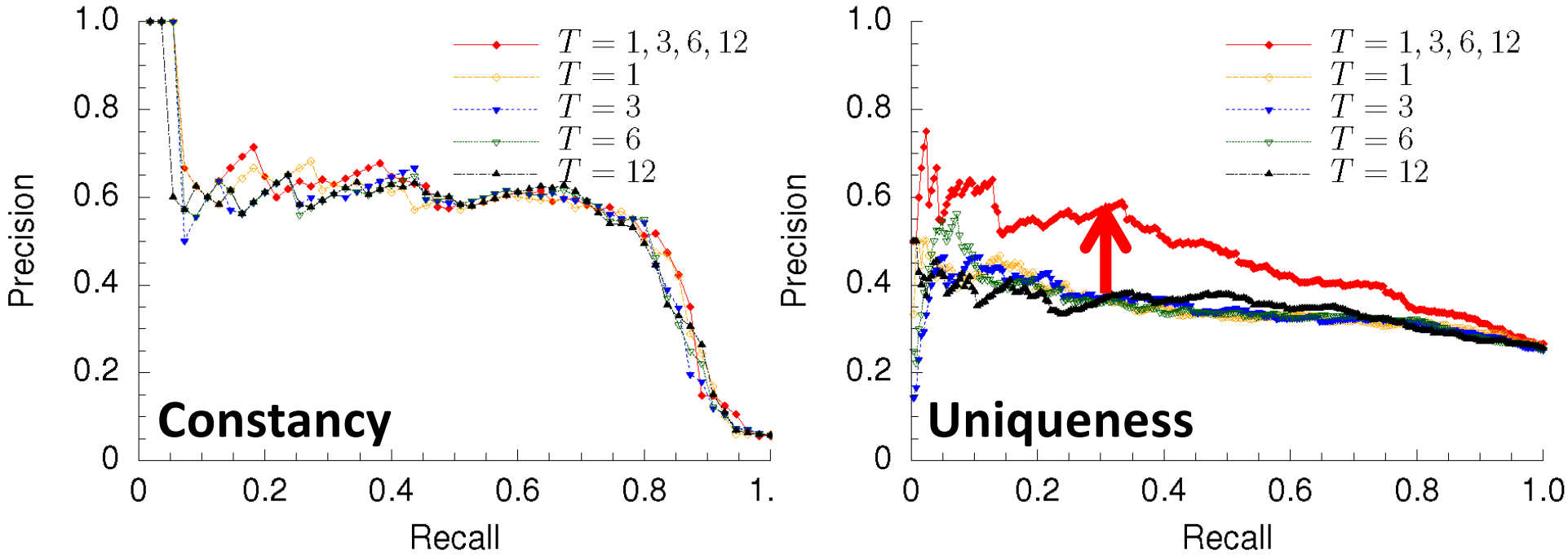
Classification Result: Uniqueness

- Varying the threshold to classifier's output (margin) to plot recall-precision curve
  - **Baseline:** re-implementation of [Lin+ 10] (based on gross distributions over arg2)



Impact of using Multiple Time Windows, *T*

- Compare our method (multiple time windows) with methods using a single time window



  - Combining features computed from multiple time windows greatly improved the precision of uniqueness classification

Error Analysis

- Investigate 200 misclassified relations

Error Type	Const.	Uniq.	Total
Paraphrases	16	52	68

ex. <**Obama**, *is president of*, **{USA, the United States}**>$_{\text{uniq.}}$

Related Work

- TempEval temporal relation identification
  [Verhagen+ 2007, 2010]
  - Associate event (relation instance) with time
    ex. <**Charles Chaplin**, *was born in*, **London**> OVERLAP 1889
  - Do not address constancy of relation

- Functional relation identification [Ritter+ 10]
  - Use distributions over arg2 to identify uniqueness [Lin+ 10]
  - Time dimension should be considered to identify uniqueness
    ex. <**Naoki Yoshinaga**, *lives in*, **{Kyoto (-'96),**
    Tokyo ('96-'05,'08-)}>$_{\textbf{uniq.}}$**

Overview

- Constancy and Uniqueness of Relations
- Our Approach
- Features for Constancy Classification
- Features for Uniqueness Classification
- Experiments
- Conclusion

Conclusion

- A novel notion of **constancy of relations**
- A method of identifying constant and unique relations
  - Use massive **time-series text to induce features**
  - Timer-series distributions over arg2 computed with multiple time windows were quite effective

- Future work
  - Apply our method to relations with *typed* arguments [Lin+ 10]
    Ex. <**arg1**$_{mountain}$, *is seen in*, **arg2**>$_{\text{const.}}$
  - Use constancy/uniqueness to compile extracted relations