

# Text-Aware Predictive Monitoring of Business Processes with LSTM Neural Networks

---

David Benedikt Georgi  
18.08.2020

Interim Presentation

## Traditional Process Mining

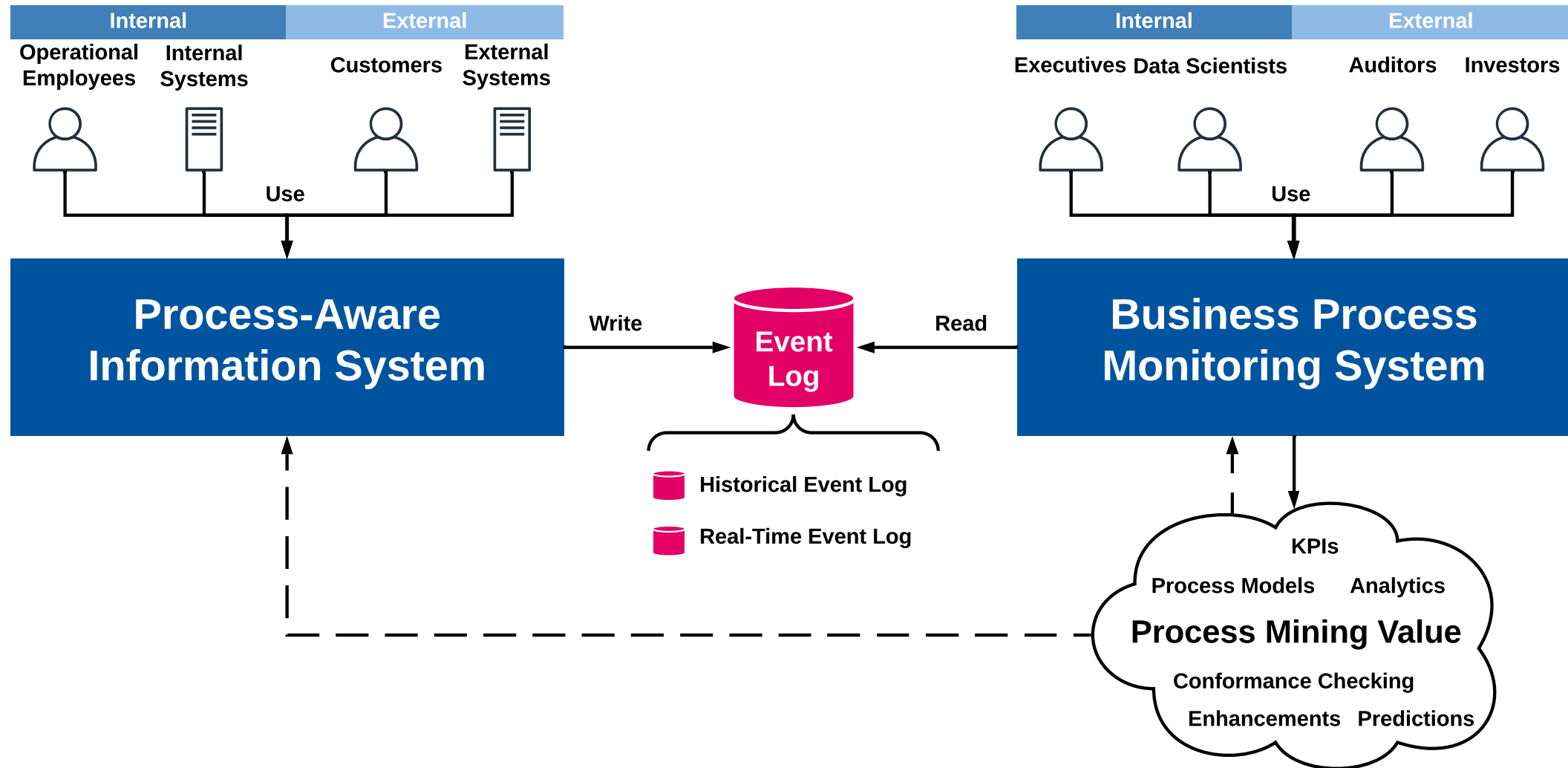
- Applied project-based
- Manual data extraction
- Historical event data only (offline)
- Standalone process mining software

VS.

## Process Mining in Business Process Monitoring

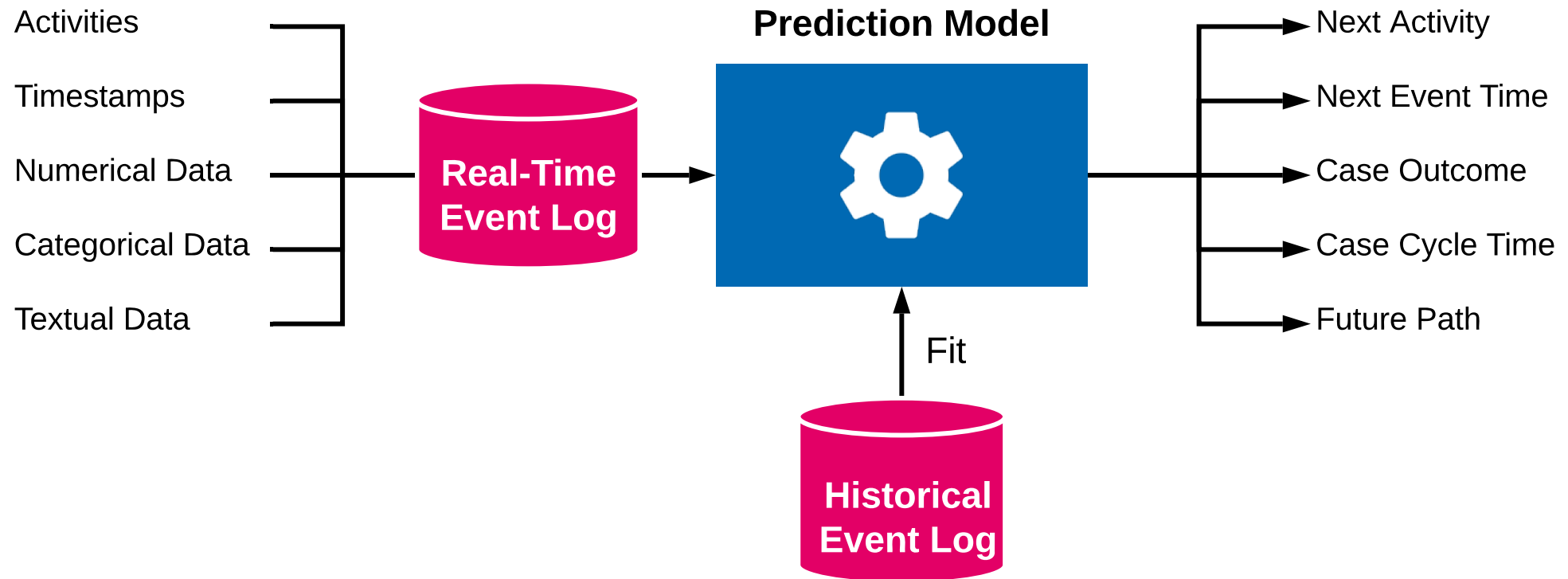
- Applied continuously
- Permanent data connection
- Historical and real-time event data (online)
- Integrated process mining platform

# Business Process Monitoring – A Modern Environment for Commercial Process Mining

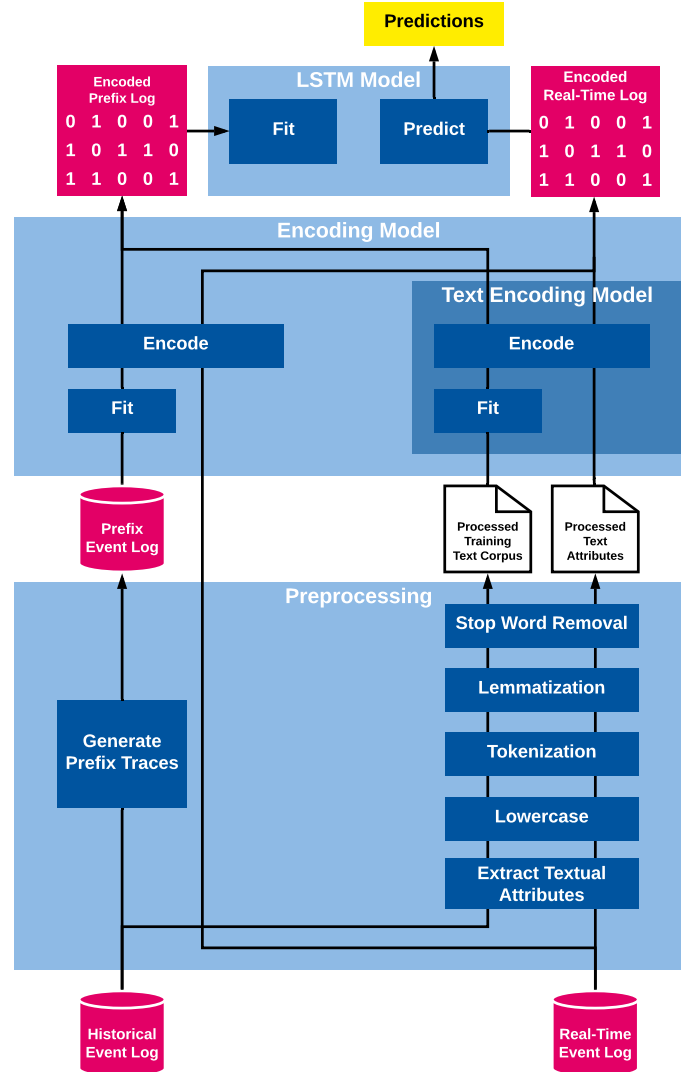


## Motivation: Adding the Forward Perspective to Business Process Monitoring

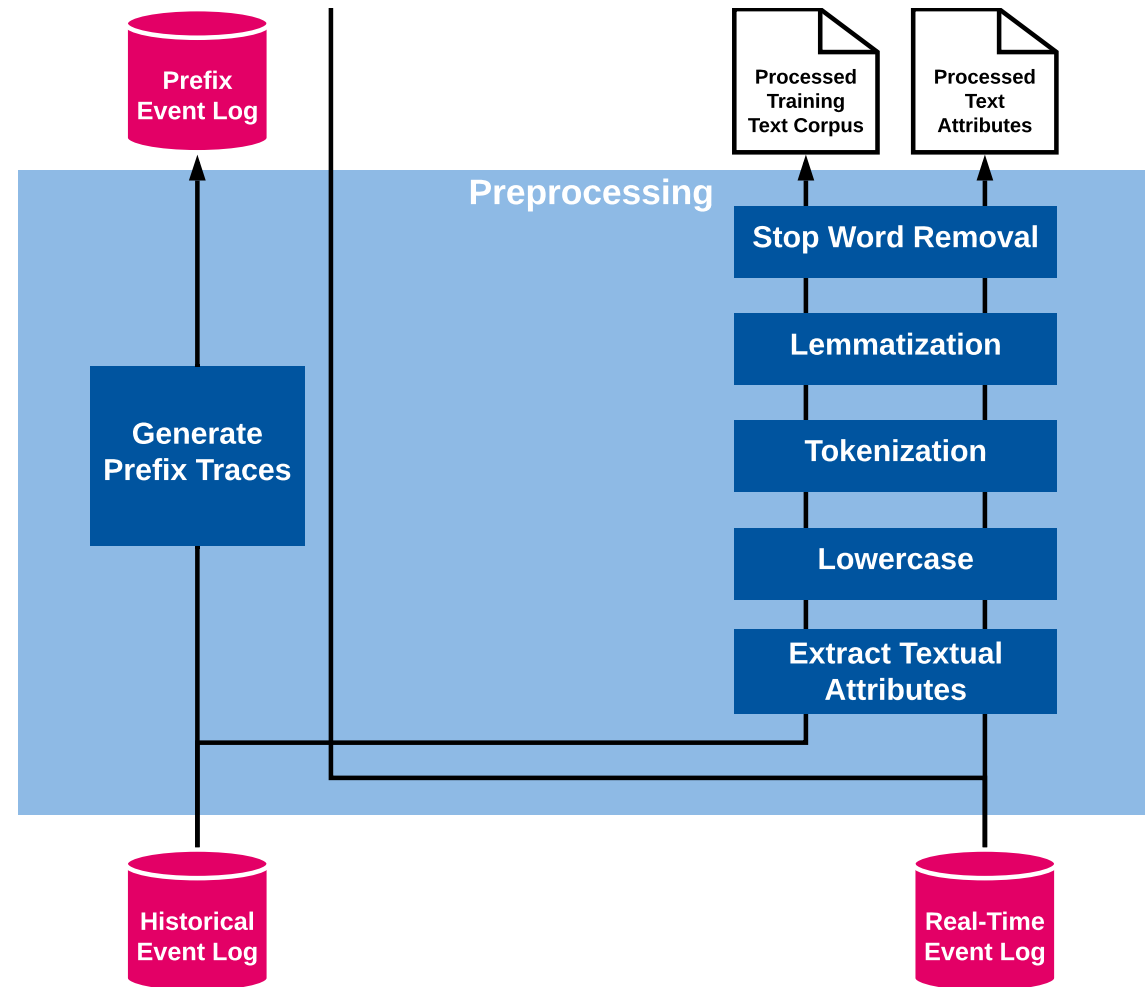
Knowing the future of a process gives organizations a competitive advantage.



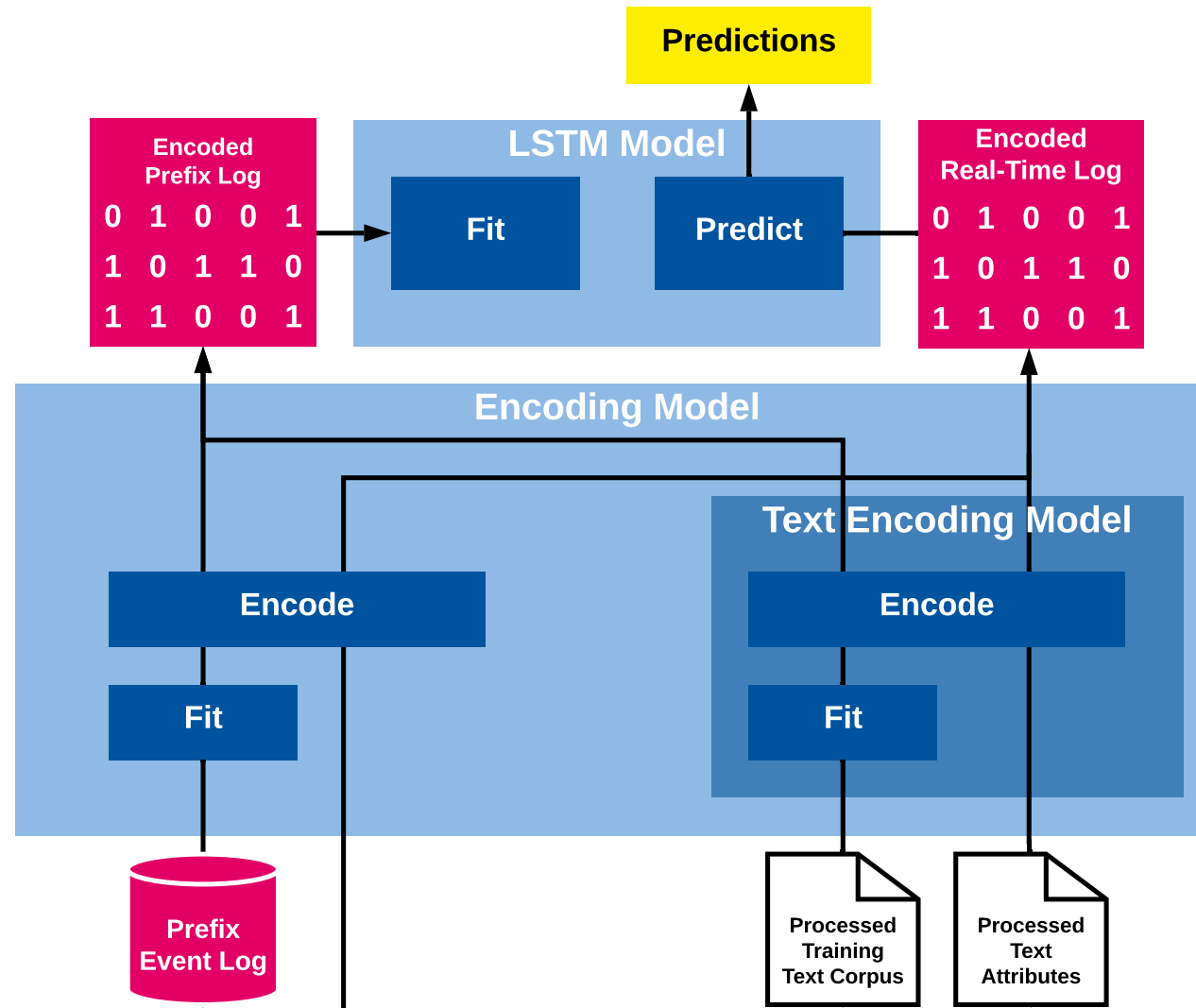
# Contribution: A Text-Aware Process Prediction Framework



# Contribution: A Text-Aware Process Prediction Framework



# Contribution: A Text-Aware Process Prediction Framework



### Transforming a sequence of events to a sequence of vectors



$e_i =$

Case ID	Activity	Timestamp	Resource	Cost	Comment
254	Consultation	02.02.2020:18.14	J. Brown, MD	67.24	"The patient has been diagnosed with high blood pressure."

$$x_i = (a_i, t_i, d_i^1, d_i^2, d_i^3)$$



# Encoding of Categorical and Numerical Attributes

## Activities & Categorical Attributes



### 1-Hot-Encoding

1-Hot(„Consultation“) = (0,1,0,0,0)

Index	Activity
1	Register patient
2	Consultation
3	Blood test
4	MRI
5	Release Patient

## Numerical Attributes



### Normalization

$$\hat{d} = \frac{d - \min(d)}{\max(d) - \min(d)}$$

## Encoding of timestamp data

- 6-dimensional vector with time-based features is part of every encoded event
- Capture daily, weekly, seasonal dependencies and concept drifts
- **Idea:** Process behavior might be influenced by office hours, weekends, seasons, etc.
- All time features are normalized to [0,1]

Feature	Description
$t_i^1$	Time since previous event
$t_i^2$	Time since case start
$t_i^3$	Time since first recorded event
$t_i^4$	Time since midnight
$t_i^5$	Time since last Monday
$t_i^6$	Time since last January 1 00:00

$$t_i = (t_i^1, t_i^2, t_i^3, t_i^4, t_i^5, t_i^6)$$

## 1. Step: Text Normalization



Transformation	Example
Original	"The patient has been diagnosed with high blood pressure."
Lowercase	"the patient has been diagnosed with high blood pressure."
Tokenization	["the", "patient", "has", "been", "diagnosed", "with", "high", "blood", "pressure", "."]
Lemmatization	["the", "patient", "have", "be", "diagnose", "with", "high", "blood", "pressure", "."]
Stop word filtering	["patient", "diagnose", "high", "blood", "pressure"]

### 2. Step: Apply Text Model

Bag of Words

Bag of N-Gram

Paragraph Vector

Latent Dirichlet Allocation

# Bag of Words

- Represent documents by the **term frequencies** (tf) of its words
- Construct **vocabulary** using the historical event log
- Create vector with size of the vocabulary with the term frequencies of the words
- Normalize each component with the **inversed document frequency** (idf)

## Example:

Vocabulary

Index	Word
1	patient
2	urgent
3	blood
4	pressure
5	notice
6	leg
7	high
8	low

Document

["patient", "diagnose", "high", "blood", "pressure"]



(1, 0, 1, 1, 0, 0, 1, 0)



(0.5, 0, 0.4, 1.2, 0, 0, 1.4, 0)

# Bag of N-Gram

- Represent documents by the **term frequencies** (tf) of its n-grams
- Construct **n-gram-vocabulary** using the historical event log
- Create vector with size of the vocabulary with the term frequencies of the words
- Normalize each component with the **inversed document frequency** (idf)

## Example: 2-Gram

Vocabulary

Index	2-Gram
1	(patient, diagnose)
2	(urgent, quick)
3	(blood, pressure)
4	(value, significant)
5	(notice, recently)
6	(leg, break)
7	(diagnose, high)
8	(high, blood)

Document

["patient", "diagnose", "high", "blood", "pressure"]



[("patient", "diagnose"), ("diagnose", "high"), ("high", "blood") and ("blood", "pressure")]

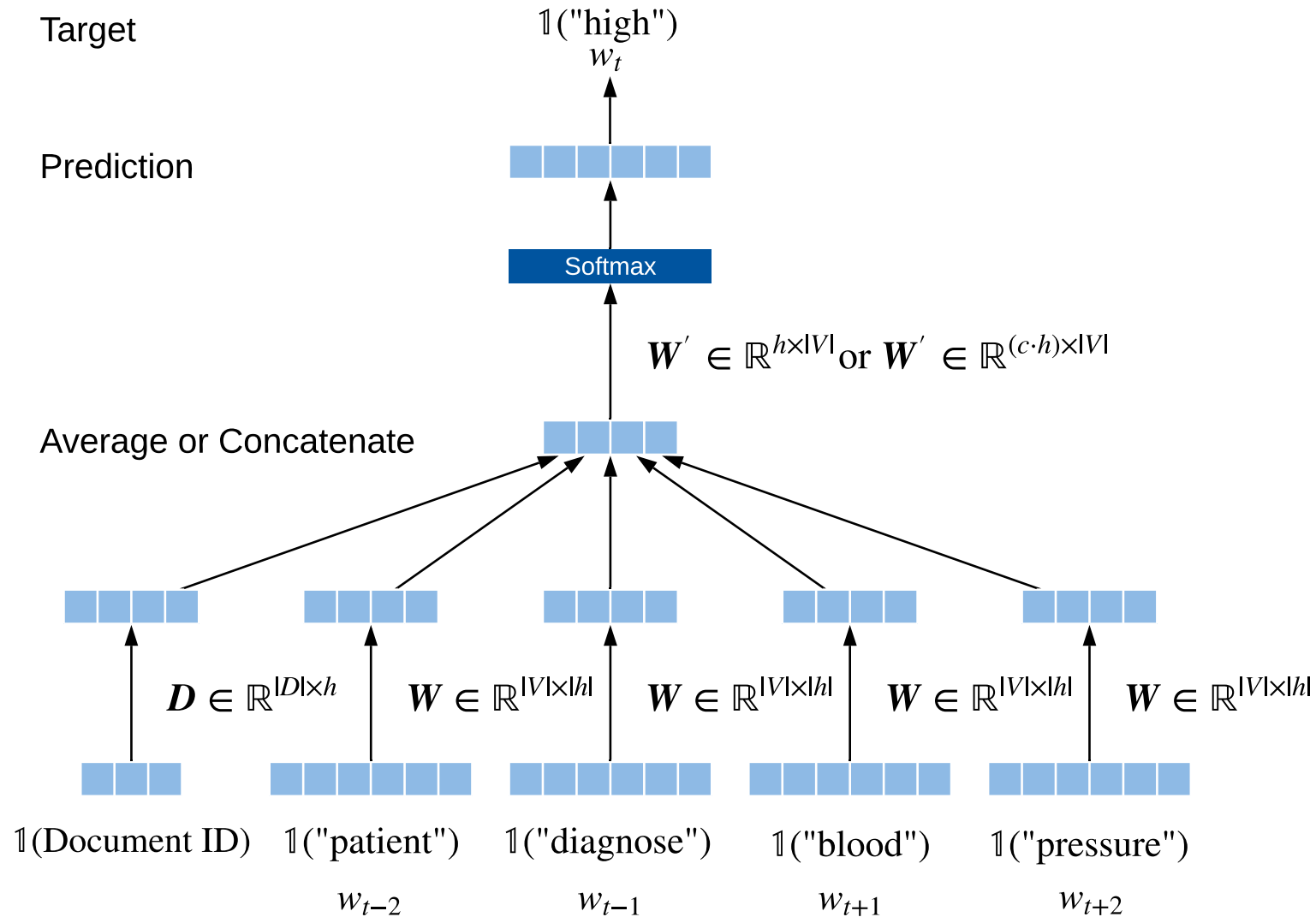


(1, 0, 1, 0, 0, 0, 1, 1)



(0.5, 0, 0.4, 0, 0, 0, 1.4, 0.6)

# Paragraph Vector



# Latent Dirichlet Allocation

## Describe documents as a probability distribution over topics

- Document is represented by a vector of dimension  $k$ , such that component  $i$  describe the “affiliation” to topic  $i$
- Topics are probability distributions over words
- Find the topic distribution for each document that most likely would generate the corresponding document

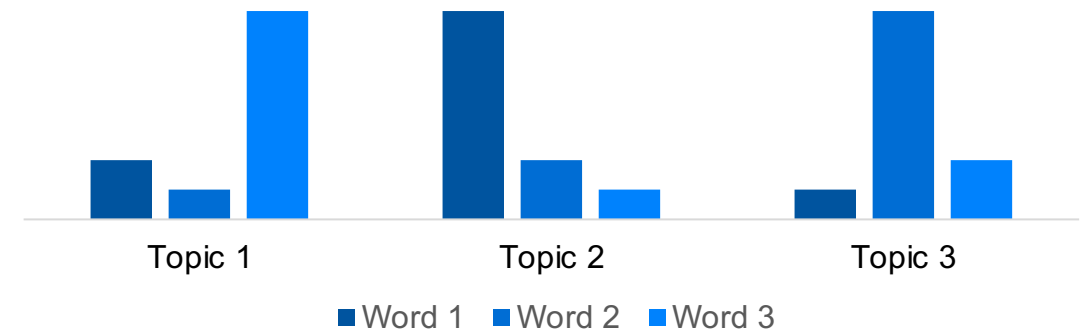
### Example: 3 Topics, 3 Words

Affiliation to topic 2

(0.1, 0.2, 0.7)

Topic distribution of a document

Topic distributions

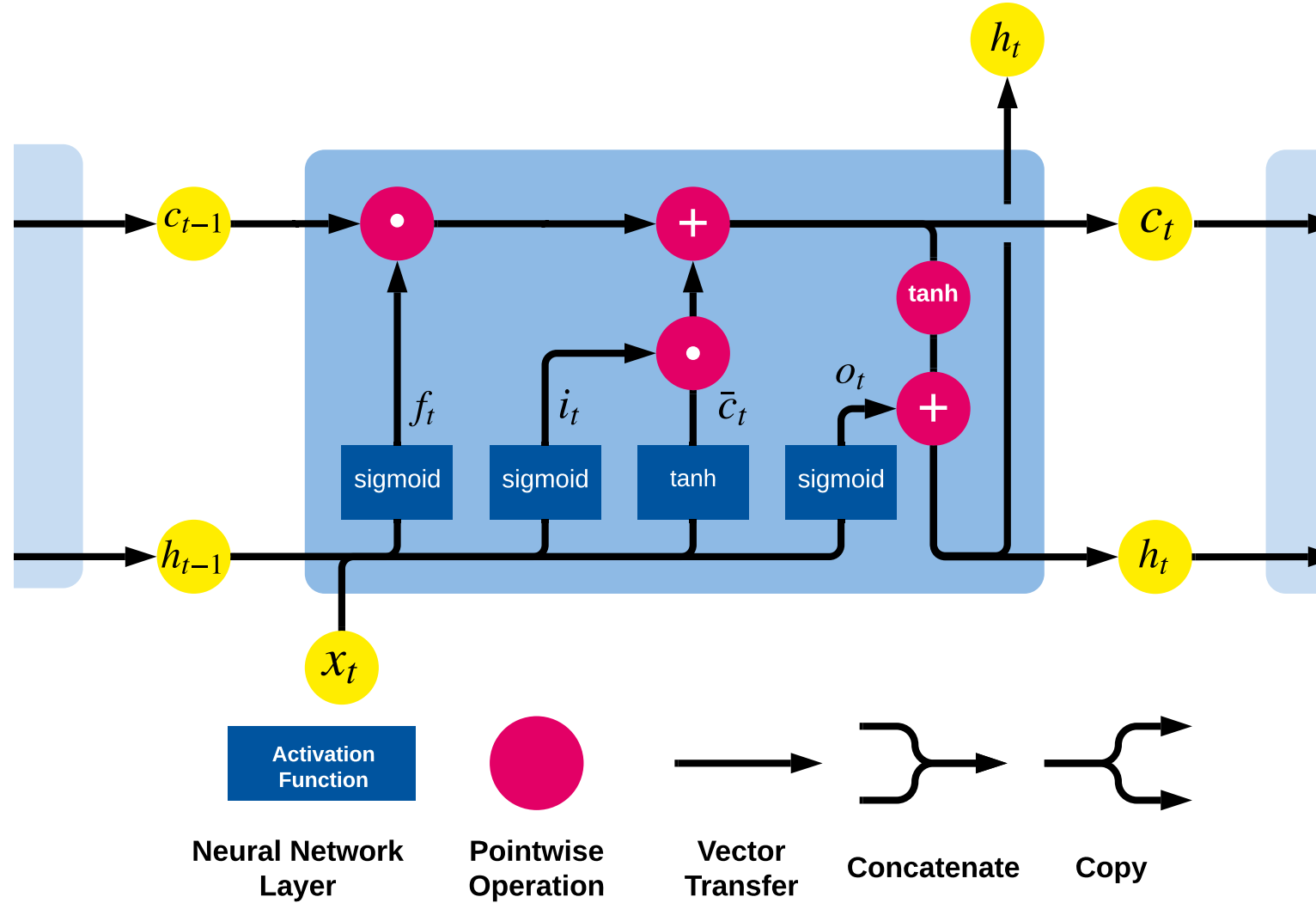




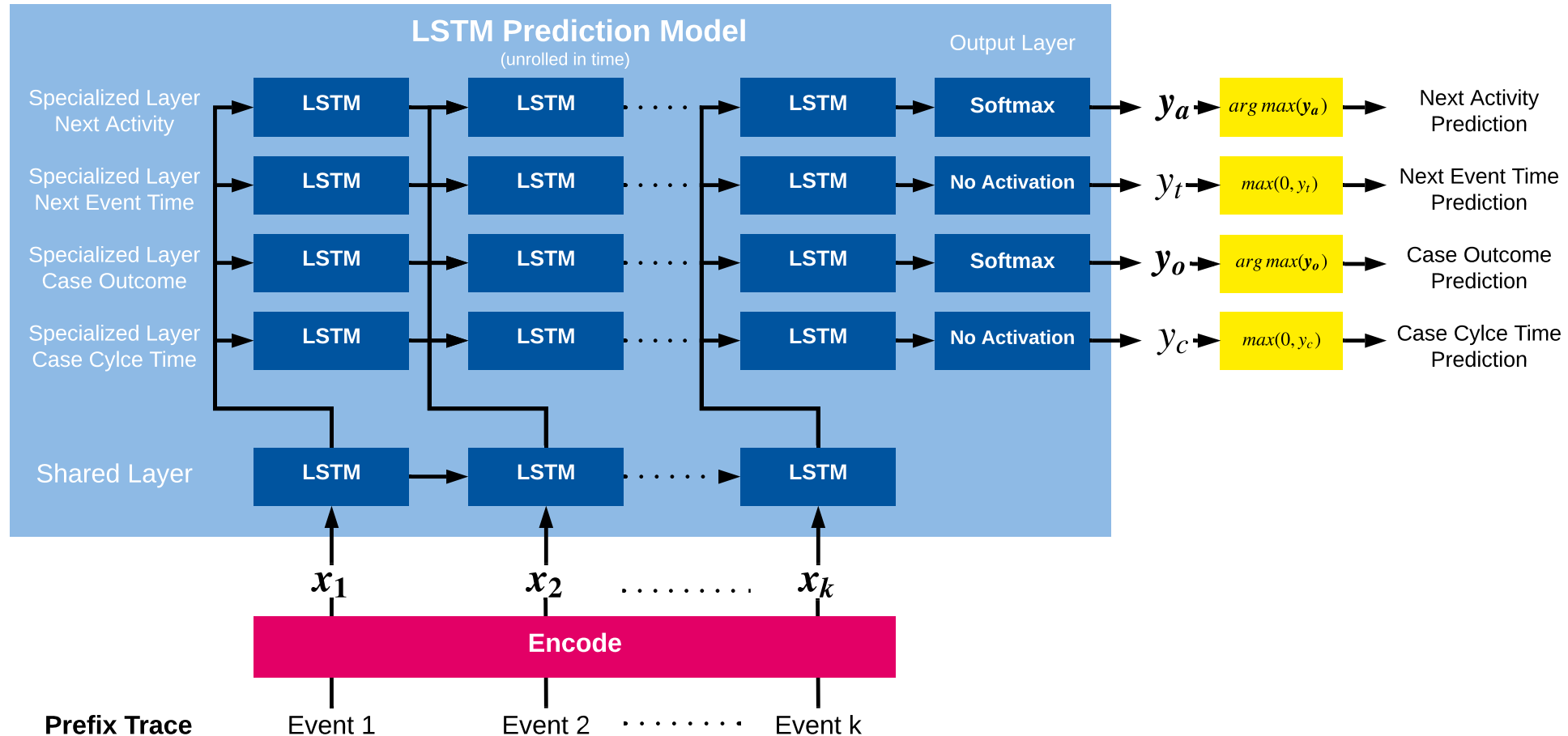
## Text Vectorization – Comparison

Text Model	Pro	Contra
Bag of Words	<ul style="list-style-type: none"><li>• Simple and fast</li></ul>	<ul style="list-style-type: none"><li>• Ignores word order</li><li>• High dimensionality</li></ul>
Bag of N-Gram	<ul style="list-style-type: none"><li>• Simple and fast</li><li>• Considers word order</li></ul>	<ul style="list-style-type: none"><li>• Very high dimensionality</li></ul>
Paragraph Vector	<ul style="list-style-type: none"><li>• Expressive representation</li><li>• Low dimensionality</li></ul>	<ul style="list-style-type: none"><li>• High computation costs</li><li>• Requires big training corpus</li></ul>
Latent Dirichlet Allocation	<ul style="list-style-type: none"><li>• Low dimensionality</li></ul>	<ul style="list-style-type: none"><li>• Ignores word order</li><li>• Choice of number of topics is difficult</li></ul>

# LSTM Module



# LSTM Network Architecture



## The Next Steps...

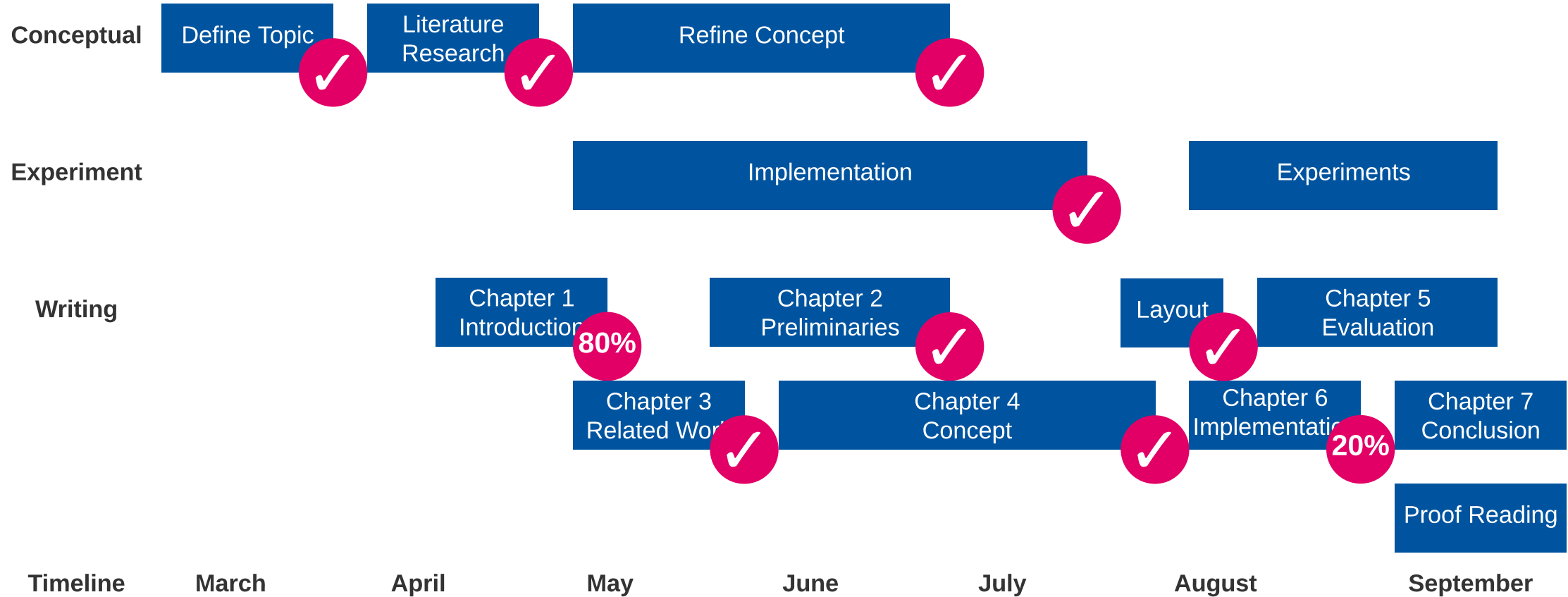
---

### Evaluation on real-life event logs

#### Research Focus:

- Impact of the utilization of textual data
- Impact of the text model choice and other parameters
- Comparison with existing techniques regarding prediction quality

# Schedule



# Thank you for your attention.

Any questions?