DSAIL

Data Science Artificial Intelligence Lab.

2023 Spring Lab Seminar

- Sequential Modelling of the Evolution of Word Representations for Semantic Change Detection (EMNLP, 2020)

2023.05.23
Jeon Hyolim
gyfla1512@g.skku.edu

Content

- 1 Introduction
- 2 Related Work
- 3 Methods
- 4 Experiments with Synthetic Data
- 5 Model Comparison with Real Data
- 6 Take home message

1 Introduction

1.1 Background

- Identifying words whose lexical meaning has changed over time is a primary
- ‘Semantic Change Detection’
  - Task: Identify words that change their meaning over time
  - Applications: historical linguistics, evolution of communities, cultural shifts …
- able to leverage the increasing availability of historical corpora in digital form and develop models that detect the shift in a word’s meaning through time

Ah, look at those blackberries, aren’t they beautiful?

Whee? I can’t see them.

There! Where this little guy is tweeting!

Huh…?

1 Introduction

1.2 Limitation of the Previous Work

1. **little work in existing literature on model comparison**
   - the lack of (longitudinal) labelled datasets,
   - -> assesses model performance mainly in a qualitative manner, without quantitative comparisons
   - difficult to assess what constitutes an appropriate approach for semantic change detection.

2. on a methodological front

- detects semantically shifted words by pairwise comparisons of their representations in distinct time periods,
  - ignoring the sequential modelling aspect of the task
  - time-sensitive process
    - considering consecutive vector representations through time : crucial

1 Introduction

- **Solution of this Work**
  - approaching semantic change detection as an anomaly identification task.
  - evolution through time via an encoder-decoder architecture
  - **hypothesize** that once such a model has been successfully trained on temporally sensitive sequences of word representations, it will accurately predict the evolution of the semantic representation of any word through time.

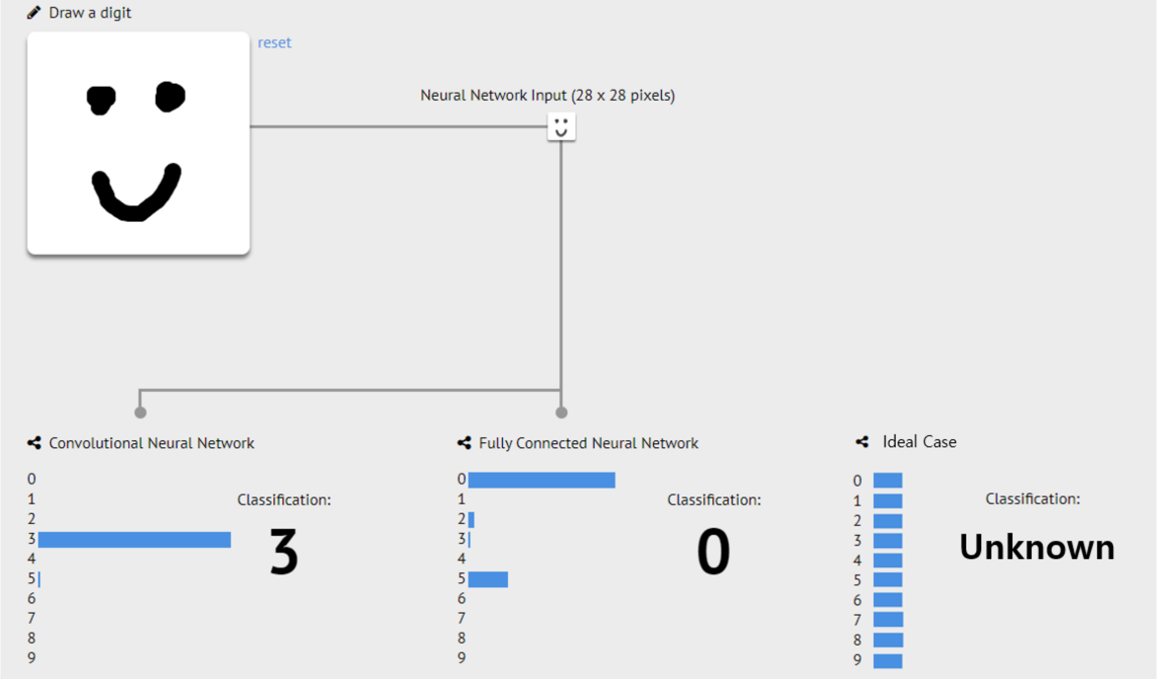
2 Related Work

2.1 Anomaly Identification Task

- **Out-of-Distribution(OOD) 문제**
  - 보유하고 있는 In-distribution 데이터셋으로 network를 학습시킨 후 Test 단계에서 ID 셋은 정확하게 예측하고 OOD 셋은 걸러내는 것
  - OOD를 Test 단계에서 어떠한 class로도 예측되지 않도록 각 class를 (1/class 개수) 확률로 균일하게 예측하는 것
  - High-confidence로 예측하지 않도록 하는 것이 목표

0 ~ 9까지의 숫자 예측하는

MINIST 관련 Task에서 '미소 짓는 얼굴' 이미지



https://hoya012.github.io/blog/anomaly-detection-overview-1/

2 Related Work

2.1 Anomaly Identification Task

- Reconstruction-based Anomaly Identification Task
  - 이상치 탐지에서 input을 복원하는 과정에서
    - **정상 데이터의 분포를 학습**하는 reconstruction 기반 방법론들이 쓰임
  - ‘정상 데이터 분포를 학습한 모델은 정상 데이터로부터 구분되는 샘플을 잘 복원하지 못할 것’이라는 가정 하에 복원 에러가 큰 샘플을 이상치로 탐지함

2 Related Work

2.2 Diachronic Representations

1. jointly learning word representations across time

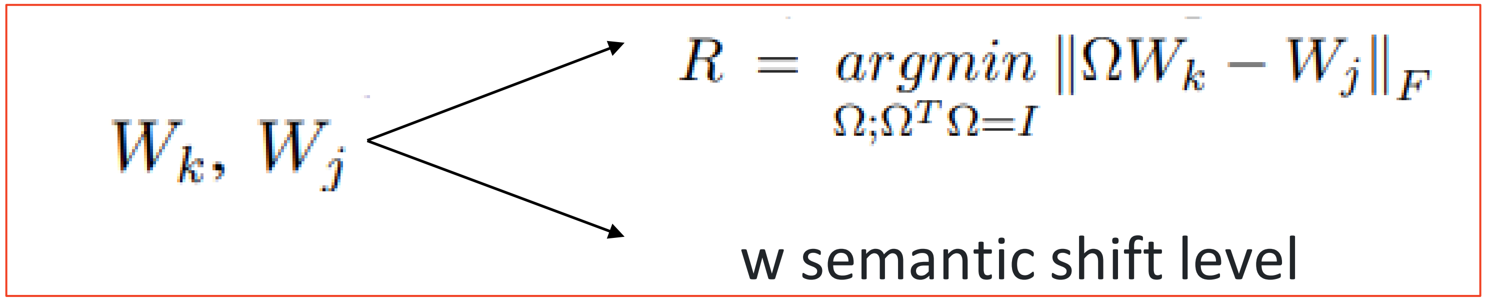
2. learning word representations over discrete time intervals (bins) and comparing the resulting vectors

- => Topics, Graphs, **Neural => word2vec**

2 Related Work

2.2 Diachronic Representations

- **Semantic change detection (Common practice)**
  - Learn word representations in two distinct time periods
  - Align them via Orthogonal Procrustes
  - Measure cosine similarity



$$W_k,\ W_j$$

$$R = \underset{\Omega;\Omega^T\Omega=I}{argmin} \|\Omega W_k - W_j\|_F$$

w semantic shift level

- Procs
  - Highly effective, Fast
- Cons
  - Strict linear transformation
  - Ignore temporality

2 Related Work

2.2 Diachronic Representations

- Approaches taking time into consideration
  - Rely on linear transformations
  - Focus on word representation

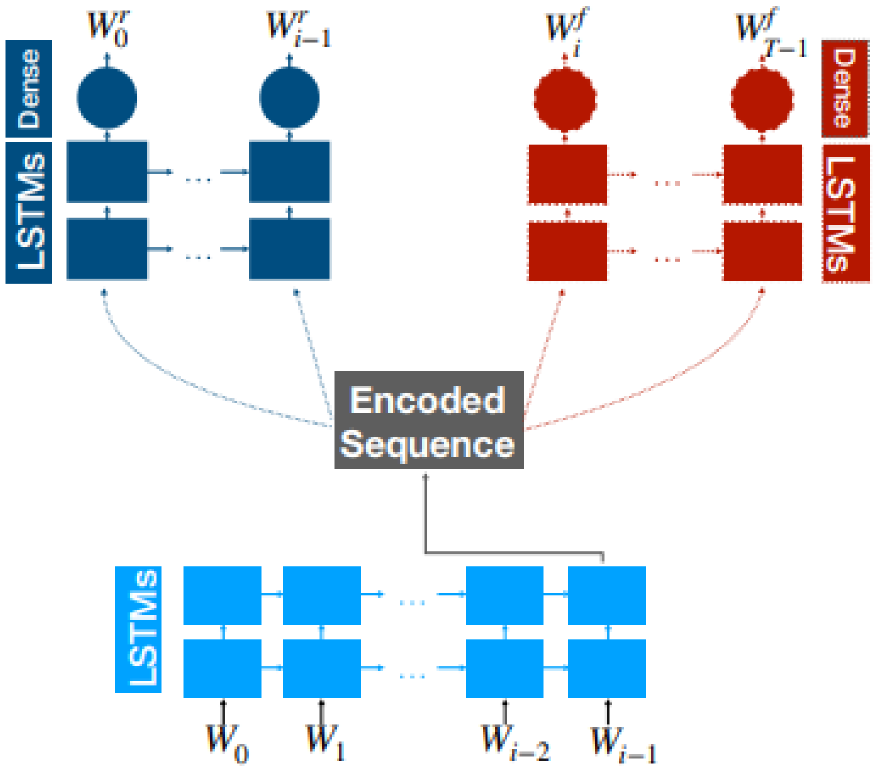
- Our contributions
  - Work with ‘any’ pre-trained word representations over time
  - Non-linear, sequential models for semantic change
  - Evaluate models in a sequential dataset, compare against strong baselines

3 Method

- Input: Pre-trained word vectors in T time periods

$$[W_0, W_1, ..., W_{i-1}, W_i, W_{i+1}, ..., W_{T-1}]$$

- Goal: Learn how the vectors evolve over time
- Semantic change: words whose sequence is hard to predict
- How
  - (a) autoencoder
  - (b) future prediction
  - (c) multi-task

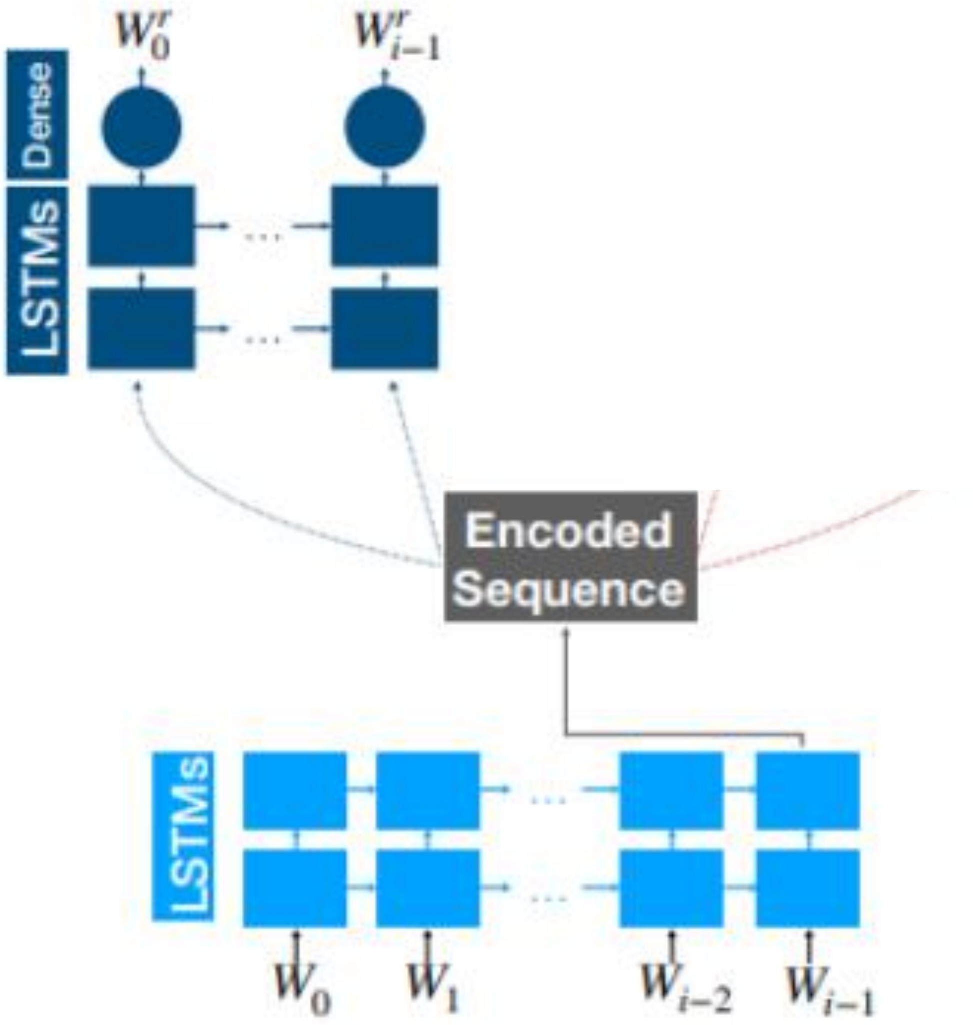


3 Method

3.1 Reconstructing Word Representations

- **(a) autoencoder (** $seq2seq_r$ **)**
  - **Reconstruct** input sequence of word vectors through time

$$W_{0:i-1} = [\underbrace{W_0, W_1, ..., W_{i-1}}]$$



$$L_r = \frac{1}{i} \sum_{j=0}^{i-1} MSE(W_j, W_j^r). \qquad (1)$$

3 Method

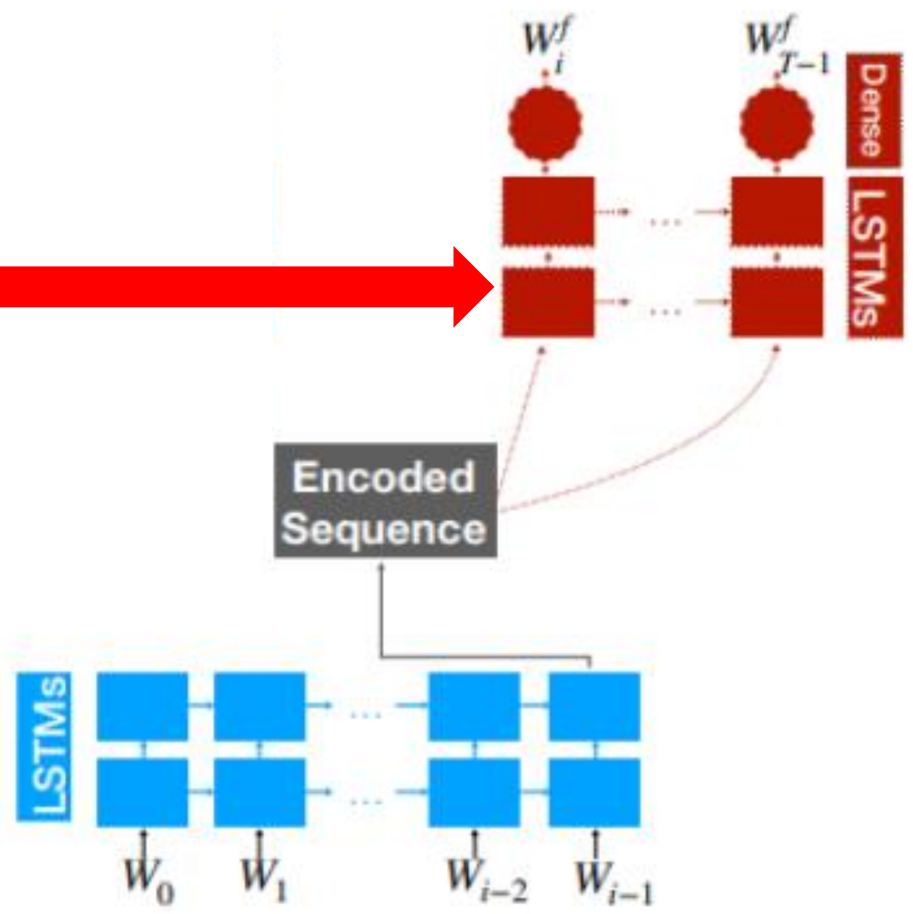
3.2 Predicting Future word Representations

- **(b) Future prediction ($seq2seq_f$)**
  - Predict future sequence of word Vectors through time

$$W_{0:i-1} = [W_0, W_1, ..., W_{i-1}]$$

사전 $W_{i:T-1} = [\overbrace{W_i, W_{i+1}, ..., W_{T-1}}]$ 에서 단어의 future representation을 예측

$$L_f = \frac{1}{T-i} \sum_{j=i}^{T-1} MSE(W_j, W_j^f). \qquad (2)$$



3 Method

3.3 Joint Model

- **(c) multi-task**
  - Reconstruct past & predict future sequences

$$W_{0:i-1} = [W_0, W_1, ..., W_{i-1}]$$

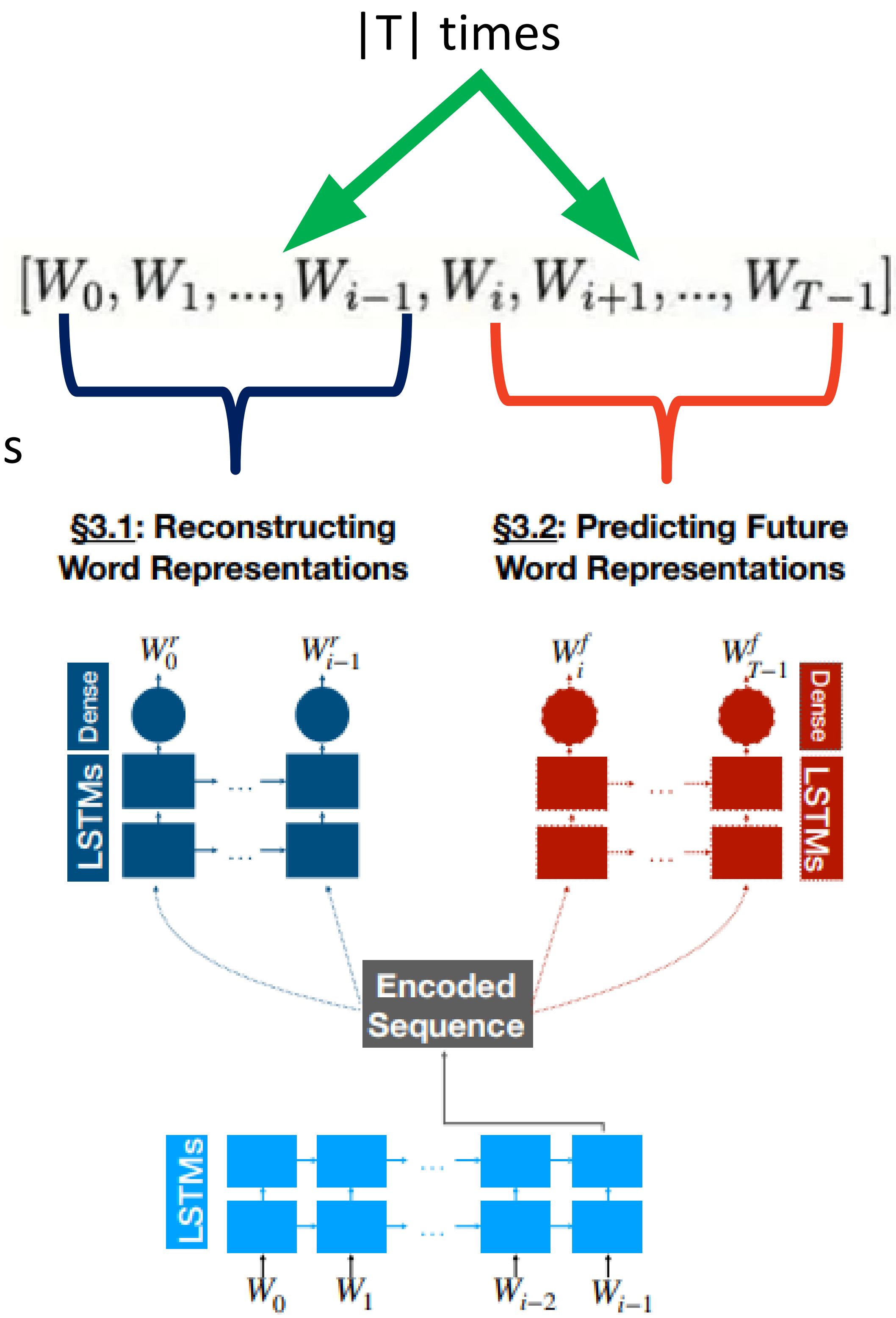
  - Goal

(1) Reconstruct the input sequence

(2) predict the future word

|T-i| representations $W_{i:T-1}$

$$L_{rf} = \frac{1}{i}\sum_{j=0}^{i-1} MSE(W_j, W_j^r) + \frac{1}{T-i}\sum_{j=i}^{T-1} MSE(W_j, W_j^f). \quad (3)$$



|T| times

$$[W_0, W_1, ..., W_{i-1}, W_i, W_{i+1}, ..., W_{T-1}]$$

§3.1: Reconstructing Word Representations

§3.2: Predicting Future Word Representations

Encoded Sequence

4 Experiments with Synthetic Data

4.1 Dataset

- UK web Archive dataset
- Size: 47.8K words
- Time period: 2000-2013
  - each year corresponds to a timestep
- Vectors: 100-dim, trained on each year independently
- Split: 80/20(train/test)

4 Experiments with Synthetic Data

4.2 Artificial Examples of Semantic Change

- Synthetic Semantic Shift: force 5% of word vectors in the test set($w_t^{(\alpha)}$) to shift their meaning towards $w_t^{(\beta)}$ over time

$$\boxed{w_t^{*(\alpha)} = \lambda_t w_t^{(\alpha)} + (1 - \lambda_t) w_t^{(\beta)}. \qquad (4)}$$

- **Conditioning on Duration of Change**

(a) "Full" [2001-13]

(b) "Half [2005-10]

(c) "OT" (One-Third) [2006-09]

(d) "Quarter" [2007-08]

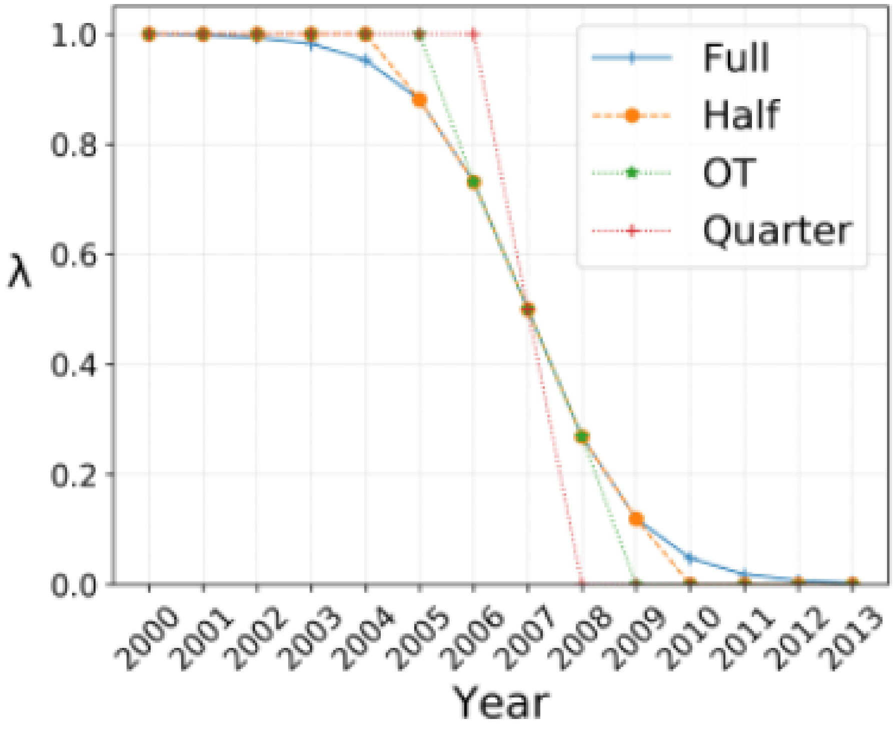


Figure 2: The different functions used to model λ_t in Eq. 4, indicating the speed and duration of semantic change of our synthetic examples (see section 4.2).

4 Experiments with Synthetic Data

4.2 Artificial Examples of Semantic Change

- Synthetic Semantic Shift: force 5% of word vectors in the test set($w_t^{(\alpha)}$) to shift their meaning towards $w_t^{(\beta)}$ over time

$$w_t^{*(\alpha)} = \lambda_t w_t^{(\alpha)} + (1 - \lambda_t) w_t^{(\beta)}. \qquad (4)$$

Conditioning on Target Words

- Bad case: {α, β} {source, target} (e.g., synonyms)
- Solution: each source word α we select uniformly at random a target word β s.t

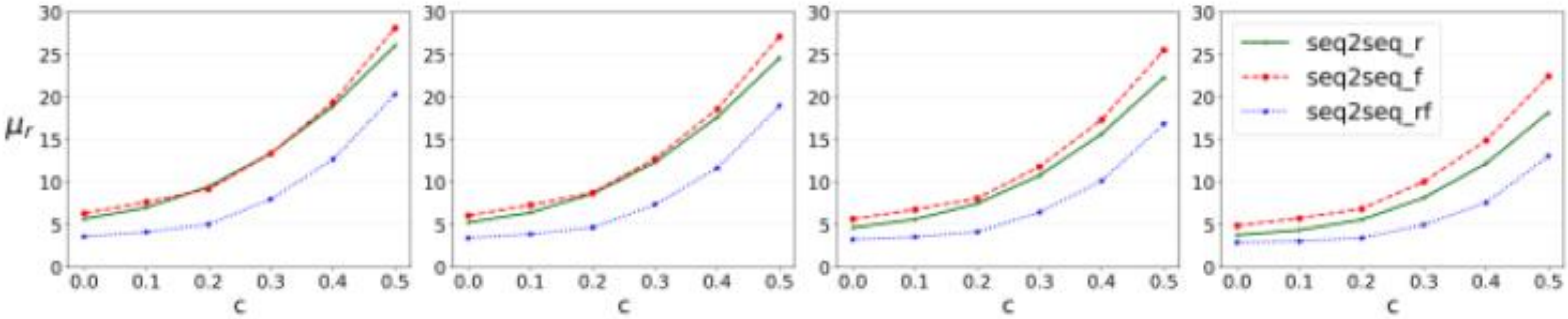
- Cosine similarity -> at the initial point => certain range: $c - 0.1 < cos(w_0^{(\alpha)}, w_0^{(\beta)}) \leq c$
- Higher values of c enforce a lower semantic change level for α through time,
- representation will be shifted towards a similar word β

- varying c = {0.0, 0.1, ..., 0.5}.

4 Experiments with Synthetic Data

4.3 Artificial Data Experiment

- **Testing and Evaluation**
  - Rank words in test set based on their average cosDist
  - High cosDist => model failure => semantic change
  - Metric: average rank of the semantically shifted words
- **Results**



(a) Full

(b) Half

(c) OT

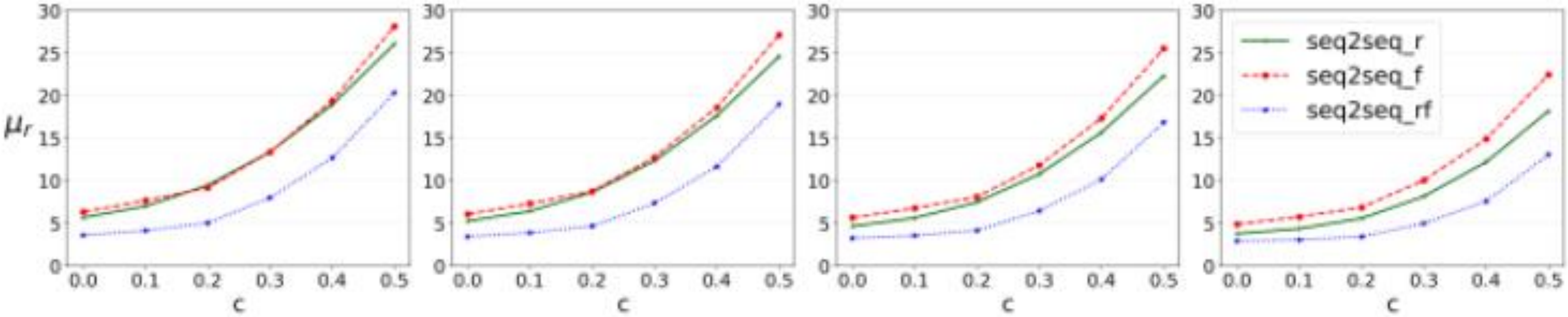
(d) Quarter

Figure 3: μ_r of our models on the synthetic dataset for different values of the threshold c (x-axis) and the different periods of duration of semantic change (one per chart, see 4.2). Lower μ_r values indicate a better performance.

4 Experiments with Synthetic Data

4.4 Results

- **Effect of conditioning Parameters**



(a) Full (b) Half (c) OT (d) Quarter

Figure 3: μ_r of our models on the synthetic dataset for different values of the threshold c (x-axis) and the different periods of duration of semantic change (one per chart, see 4.2). Lower μ_r values indicate a better performance.

5 Model Comparison with Real Data

5.1 Experimental Setting

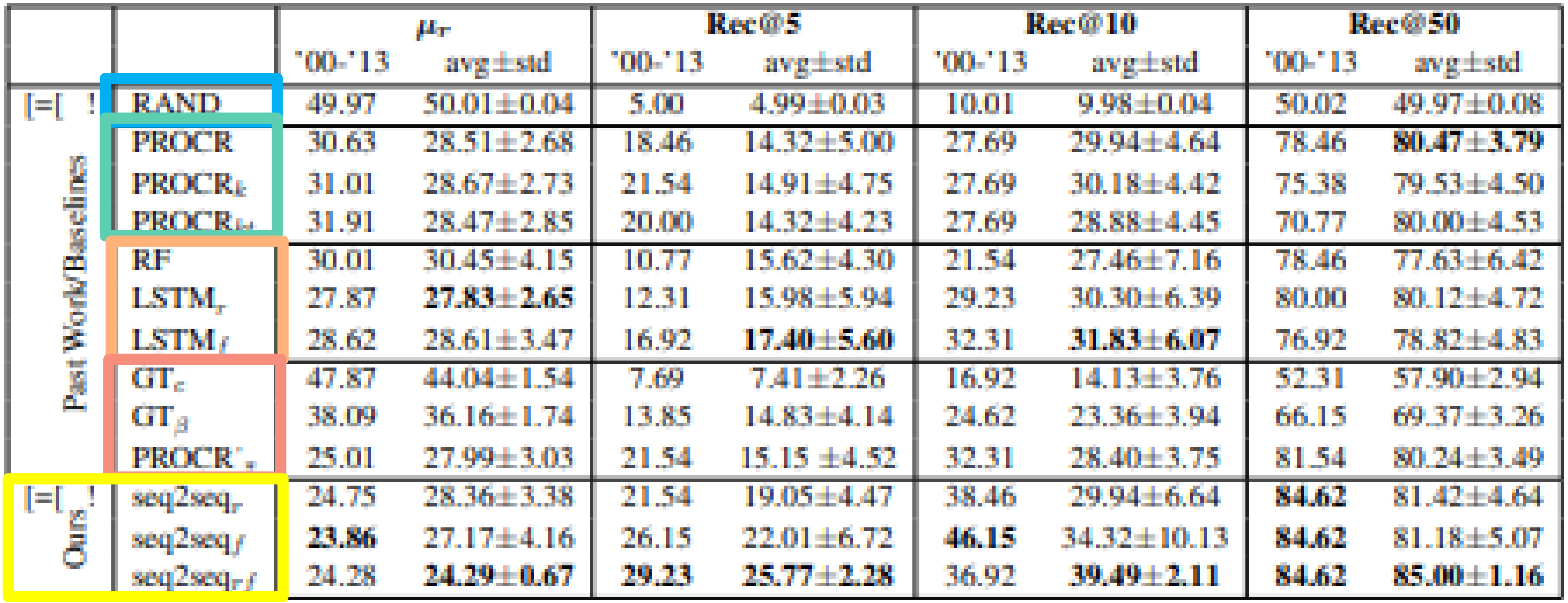
- **Data**: 47.8 k words → 65 words with altered meaning
- **Goal:** Better rank for the 65 words with altered lexical semantics
- **Metrics**: μr, Rec@k scores

- **Models**
  - A random word rank generator (RAND)
  - Variants of Procrustes Alignment
  - Models leveraging the first and last word representations only
  - Models operating on the time series of distances.

5 Model Comparison with Real Data

5.2 Results

- **Our model vs baselines**



		μ_r		Rec@5		Rec@10		Rec@50	
		'00-'13	avg±std	'00-'13	avg±std	'00-'13	avg±std	'00-'13	avg±std
[=[!	RAND	49.97	50.01±0.04	5.00	4.99±0.03	10.01	9.98±0.04	50.02	49.97±0.08
Past Work/Baselines	PROCR	30.63	28.51±2.68	18.46	14.32±5.00	27.69	29.94±4.64	78.46	**80.47±3.79**
	PROCR$_k$	31.01	28.67±2.73	21.54	14.91±4.75	27.69	30.18±4.42	75.38	79.53±4.50
	PROCR$_{kt}$	31.91	28.47±2.85	20.00	14.32±4.23	27.69	28.88±4.45	70.77	80.00±4.53
	RF	30.01	30.45±4.15	10.77	15.62±4.30	21.54	27.46±7.16	78.46	77.63±6.42
	LSTM$_r$	27.87	**27.83±2.65**	12.31	15.98±5.94	29.23	30.30±6.39	80.00	80.12±4.72
	LSTM$_f$	28.62	28.61±3.47	16.92	**17.40±5.60**	32.31	**31.83±6.07**	76.92	78.82±4.83
	GT$_c$	47.87	44.04±1.54	7.69	7.41±2.26	16.92	14.13±3.76	52.31	57.90±2.94
	GT$_\beta$	38.09	36.16±1.74	13.85	14.83±4.14	24.62	23.36±3.94	66.15	69.37±3.26
	PROCR'$_a$	25.01	27.99±3.03	21.54	15.15 ±4.52	32.31	28.40±3.75	81.54	80.24±3.49
[=[! Ours	seq2seq$_r$	24.75	28.36±3.38	21.54	19.05±4.47	38.46	29.94±6.64	**84.62**	81.42±4.64
	seq2seq$_f$	**23.86**	27.17±4.16	26.15	22.01±6.72	**46.15**	34.32±10.13	**84.62**	81.18±5.07
	seq2seq$_{r,f}$	24.28	**24.29±0.67**	**29.23**	**25.77±2.28**	36.92	**39.49±2.11**	**84.62**	**85.00±1.16**

Table 1: Model comparison when operating on the entire time sequence (2000-13) and averaged across time (2000-01, ..., 2000-13). Past work and baseline models shown in the table are defined in section 5.1 ("Models").

5 Model Comparison with Real Data

5.2 Results

- **Effect of input/output lengths**

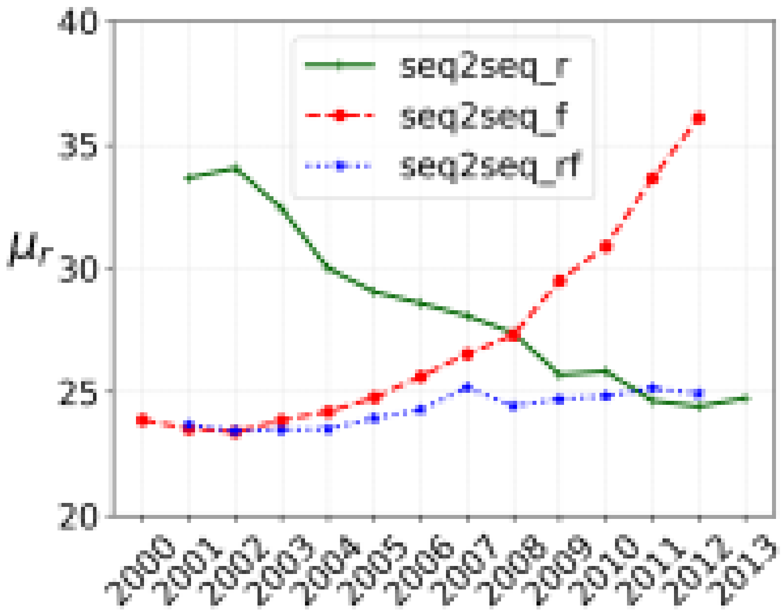


Figure 5: μ_r of our models for varying value of i (Eq. 1–3).[5]For the complete results, refer to Appendix B.

- **Words with shifted meaning**

Parmesan

Even

unlike

tweet

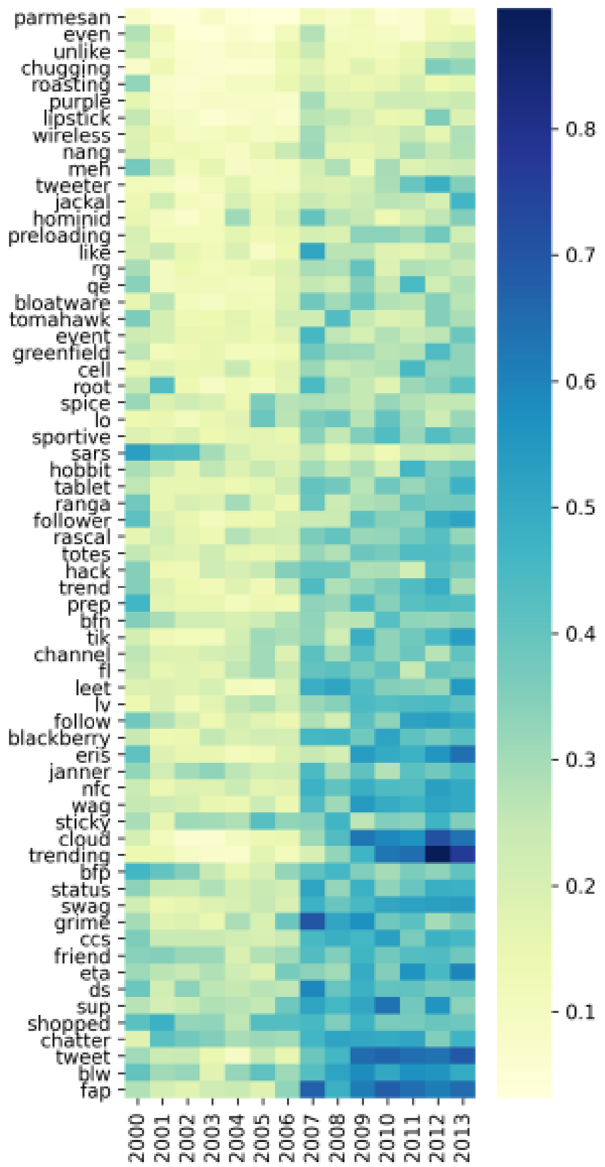


Figure 6: Cosine distances (actual *vs* predicted vectors) of each semantically shifted word (as indicated by the Oxford English Dictionary), per year. Lighter colours indicate better model performance – thus, lower level of semantic change predicted by our joint model.

6 Take home message

6.1 Conclusion

- LSTM 기반 아키텍처에서 3 variants of non-linear + sequential model
- 단어의 semantic change의 level을 측정하기 위해서
  - sequential한 방법으로 time 별로 evolution을 tracking함:
    - (a) word representation autoencoder,
    - (b) future word representation decoder
    - (c) (a), (b)를 결합한 하이브리드한 방법
- synthetic data에서 실험해서 모델의 효율성 입증
- 실제 데이터를 사용해서 베이스라인에서 경쟁력을 비교
  - 성능 향상을 입증
  - 시간 별로의 word vectors의 sequential modelling의 중요성을 강조함

6 Take home message

6.2 Future work(Types of Out-of-Distribution Texts and How to Detect Them (EMNLP, 2021)

- NLP의 경우, **일반적인 분포 변화의 유형에 따라 OOD 예제를 분류**
- 구체적으로, 입력(예: 영화 리뷰)은
  - 1) 서로 다른 레이블에서 불변인 **background features**(예: 장르)과
  - 2) 예측 작업에서 구별력이 있는 **semantic features**(예: 감성 단어)으로
  - 표현될 수 있다고 가정
- test time 때 **주요한 변화로 특징화 되는 두 가지 유형의 OOD 예제를 고려**
  - background와 semantic features
- 종종 동시에 발생하지만, 이런 경우 하나가 우세함
  - 예1) background feature가 우세: 도메인이나 텍스트의 스타일이 변경되는 경우
  - 예2) semantic feature가 우세: test 때 unseen classes 가 등장한 경우

6 Take home message

6.2 Future work(Types of Out-of-Distribution Texts and How to Detect Them (EMNLP, 2021)

- **Method**
- 두 분류(background, semantic features)를 사용해
- OOD 탐지의 두 가지 주요 접근 방식-> 평가
    - **모델의 예측 신뢰도를 사용**하는 보정(calibration) 방법 <= semantic shift
    - **학습 입력의 분포를 적합하**는 밀도 추정(density estimation) 방법 <= background shift

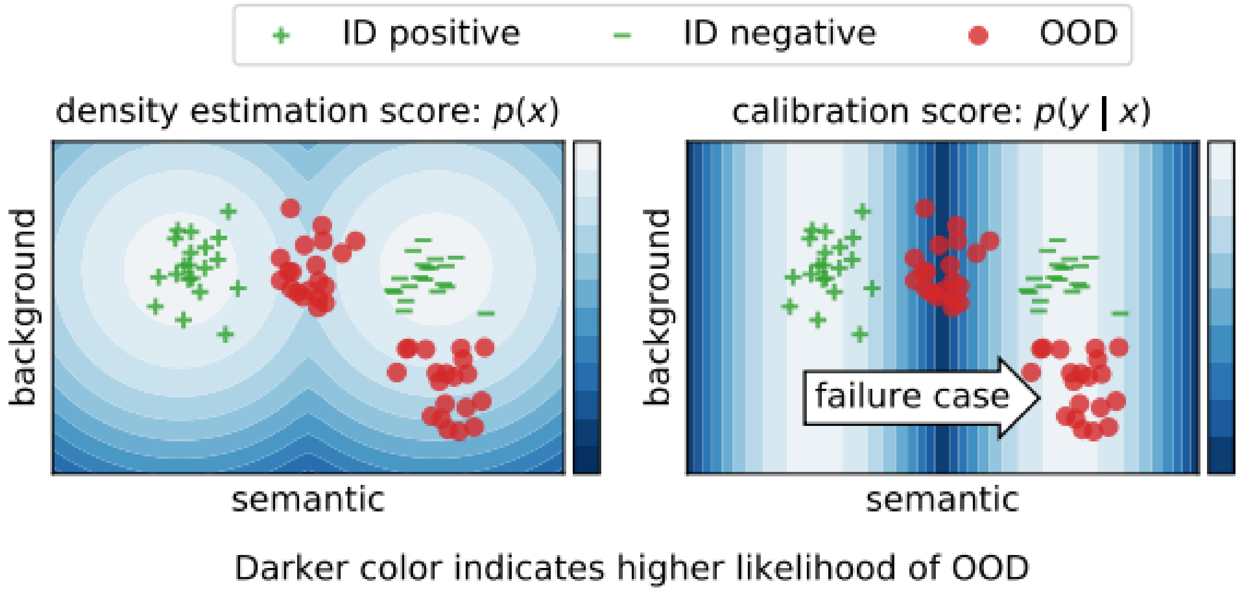


Figure 1: Illustration of semantic shift and background shift in $\mathbb{R}^2$. Each point consists of semantic features (x-axis) and background features (y-axis). OOD examples (red points) can shift in either direction. The background color indicates regions of ID (light) and OOD (dark) given by the density estimation method (left) and the calibration method (right). The calibration method fails to detect OOD examples due to background shift.

6 Take home message

6.2 Future work

- Video(Paper talk)
  - https://papertalk.org/papertalks/11741
- Sequential Modelling of the Evolution of Word Representations for Semantic Change Detection(EMNLP, 2020)
  - https://velog.io/@gyfla1512/NLP-23-1-Sequential-Modelling-of-the-Evolution-of-Word-Representations-for-Semantic-Change-DetectionEMNLP-2020
- Types of Out-of-Distribution Texts and How to Detect Them(EMNLP, 2021)
  - https://velog.io/@gyfla1512/NLP-23-1-Types-of-Out-of-Distribution-Texts-and-How-to-Detect-Them

DSAIL

Data Science Artificial Intelligence Lab.

A. Appendix

A.1 Orthogonal Procrustes problem

- 행렬대수학에서 쓰이는 matrix approximation 문제
  - 두 행렬에 주어지면 가장 잘 map하는 orthogonal matrix를 찾고자 함

$$R = \arg\min_{\Omega} \|\Omega A - B\|_F \quad \text{subject to} \quad \Omega^T \Omega = I,$$

- 증명

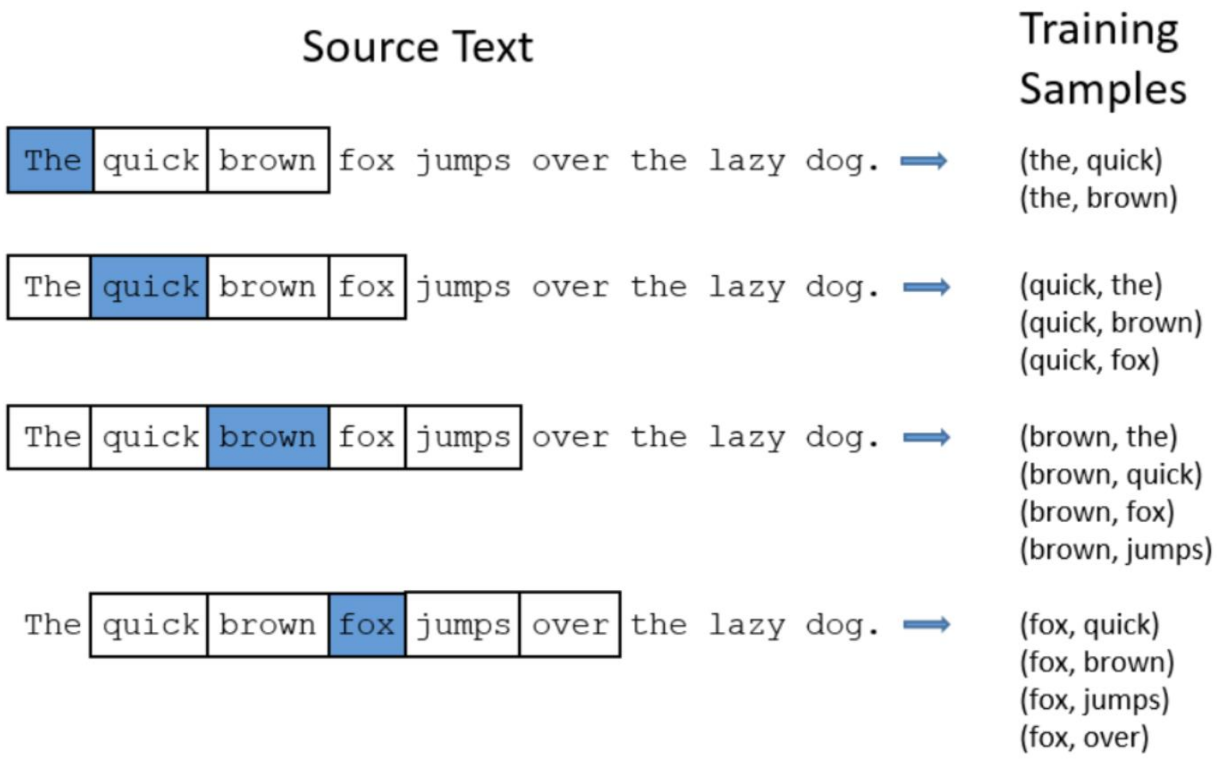
$$\begin{aligned}
R &= \arg\min_{\Omega} ||\Omega A - B||_F^2 \\
&= \arg\min_{\Omega} \langle \Omega A - B, \Omega A - B \rangle_F \\
&= \arg\min_{\Omega} ||\Omega A||_F^2 + ||B||_F^2 - 2\langle \Omega A, B \rangle_F \\
&= \arg\min_{\Omega} ||A||_F^2 + ||B||_F^2 - 2\langle \Omega A, B \rangle_F \\
&= \arg\max_{\Omega} \langle \Omega, BA^T \rangle_F \\
&= \arg\max_{\Omega} \langle \Omega, U\Sigma V^T \rangle_F \\
&= \arg\max_{\Omega} \langle U^T \Omega V, \Sigma \rangle_F \\
&= \arg\max_{\Omega} \langle S, \Sigma \rangle_F \quad \text{where } S = U^T \Omega V
\end{aligned}$$

https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem

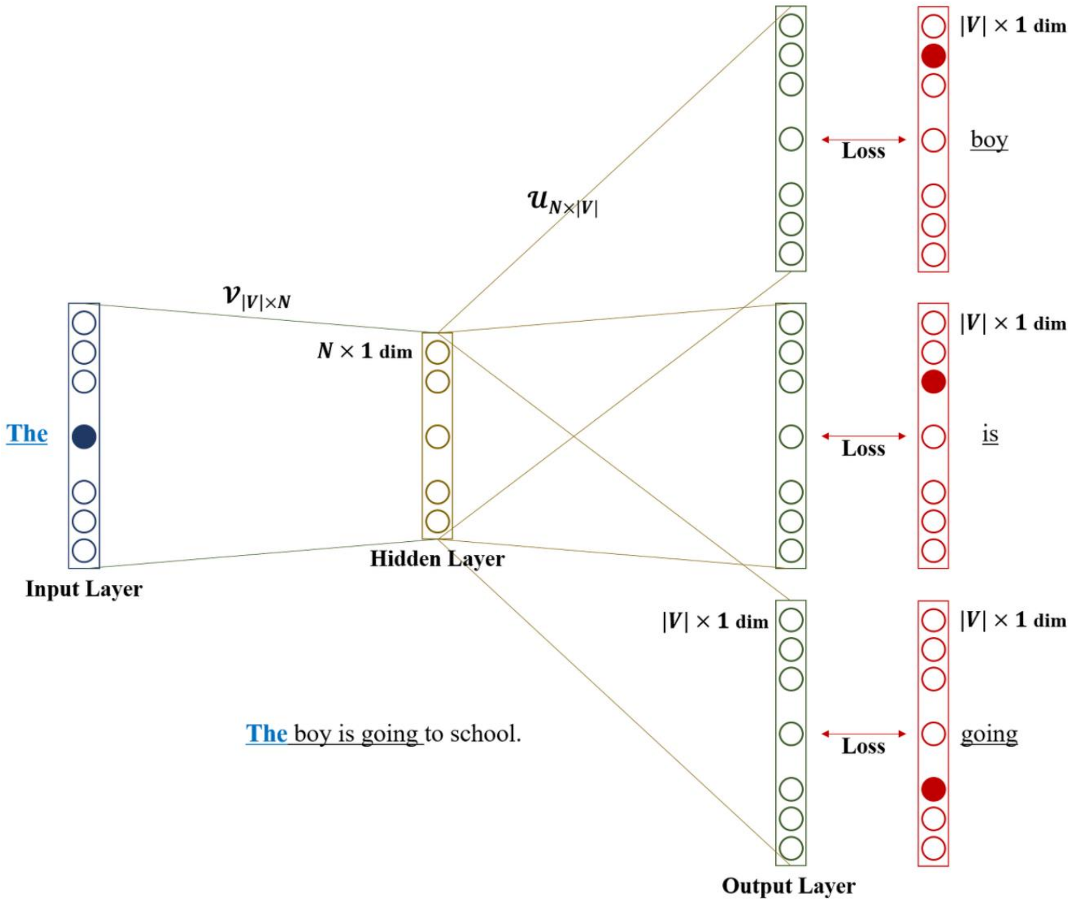
A. Appendix

A.2 Skip-Gram

- 문맥이 주어지면 기준 단어로부터 문맥 단어를 예측하는 모델
  - 기준 단어를 입력으로 사용해서 기준 단어에 대해 앞 뒤로 N/2개 씩, 총 N개의 문맥 단어를 맞추기 위한 네트워크를 생성함



Source Text	Training Samples
The quick brown fox jumps over the lazy dog. ⟹	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. ⟹	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. ⟹	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. ⟹	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

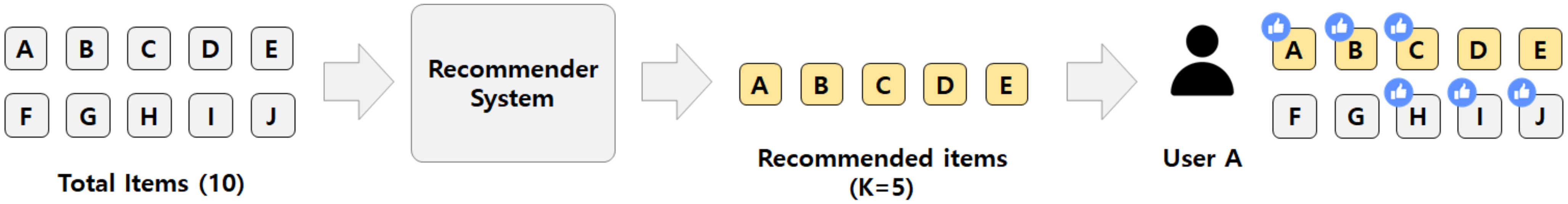


https://wooono.tistory.com/244

A. Appendix

A.2 Recall @ k

- 추천시스템에서 하나가 아니라 여러 개를 추천하는 경우
  - 정답도 여러 개가 됨(k는 추천 아이템 수)
- Recall @ k: 사용자가 관심 있는 모든 아이템 중에서 내가 추천한 아이템 k개가 얼마나 포함 되는지의 비율



$$Precision@5 = \frac{3}{5} = 0.6 \qquad Recall@5 = \frac{3}{6} = 0.5$$

https://sungkee-book.tistory.com/11#:~:text=Recall%40K%EB%8A%94%20%EC%82%AC%EC%9A%A9%EC%9E%90%EA%B0%80,%405%3D0.5%EA%B0%80%20%EB%90%9C%EB%8B%A4.