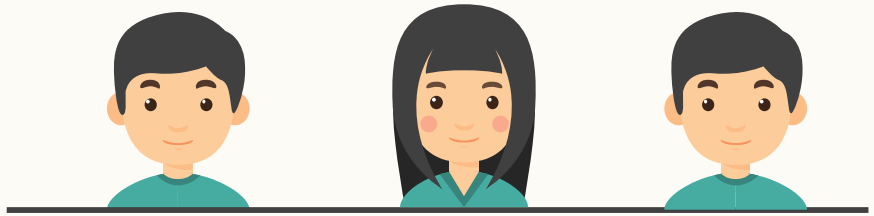
LEAGUE OF LEGENDS

실시간 채팅을 이용한 승패 예측

비정형 데이터 분석 최종 발표



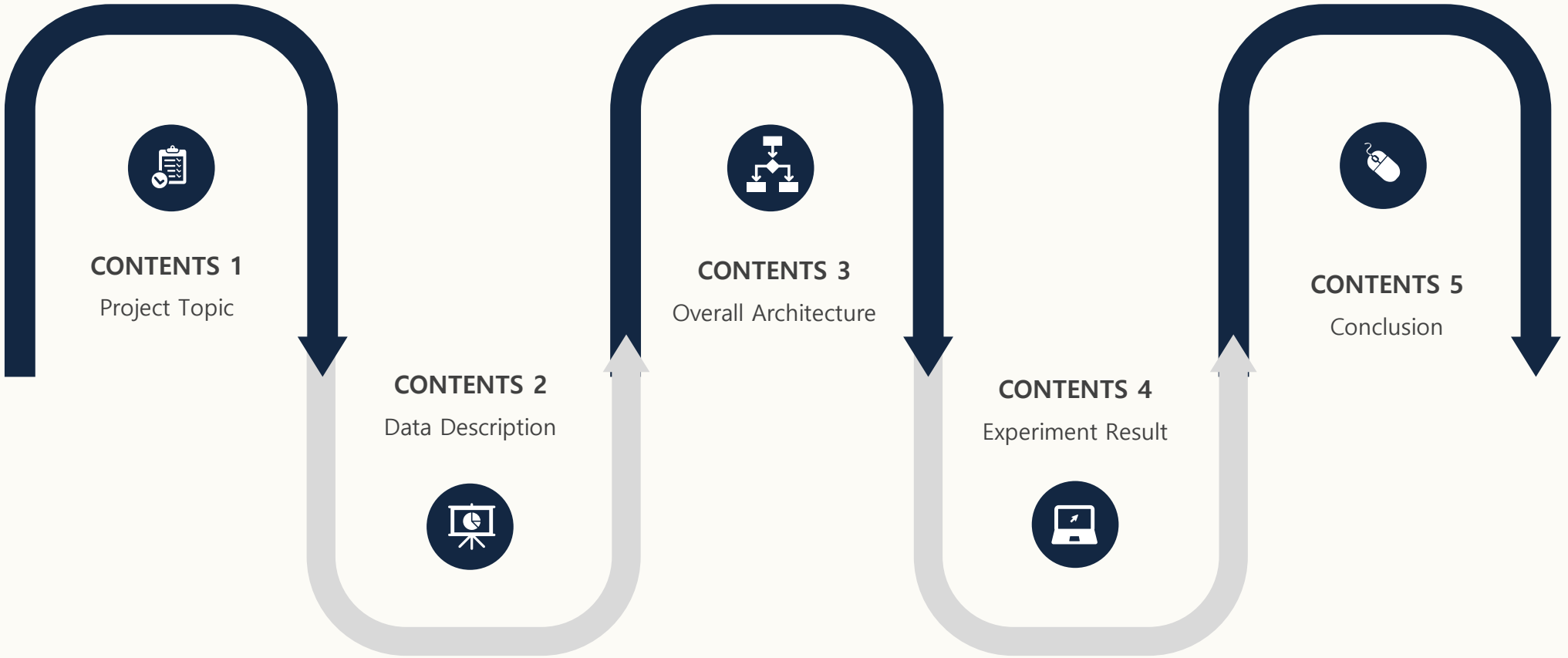
김두형
2020020544

강민지
2020010557

이용택
2020020551

Contents

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!



Project Topic

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Motivation
  - 우리나라를 포함하여 전 세계적으로 E-sports 시장의 성장세를 보이고 있음
  - E-sports 중에서 최근 세계적으로 시장을 이끌고 있는 게임이 Riot Games Inc.의 League of Legends임
  - 주기적으로 세계 각지에서 열리는 대회를 보기 위한 스트리밍 시장이 같이 성장하고 있음
  - 가장 큰 스트리밍 플랫폼이 Twitch임

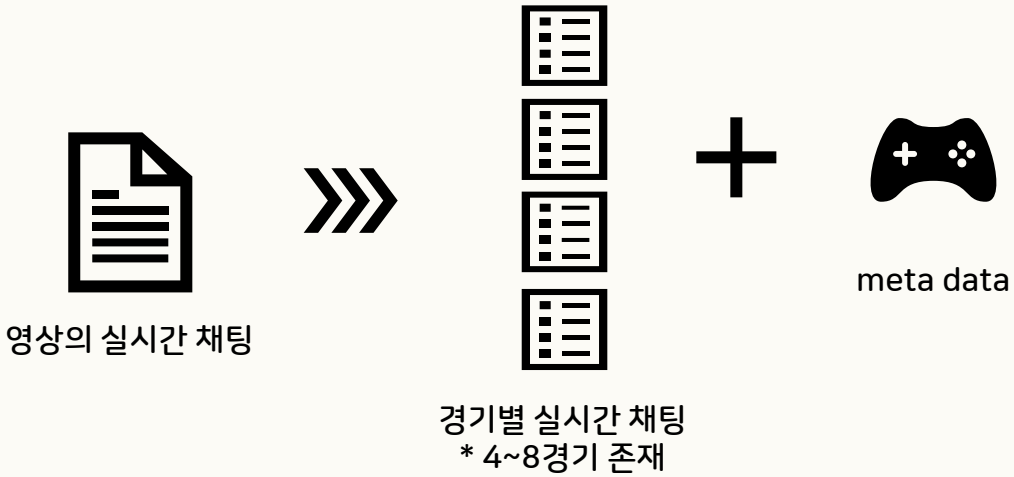
스트리밍 시장의 꽃이라 불리는 실시간 채팅(text)이 영상을 잘 반영하고 있을지 의문

- Purpose
  - 실시간 채팅에 특정 선수와 캐릭터를 언급하면서 전반적인 게임의 내용을 대화하기 때문에 게임의 진행상황이 반영되었을 것이라고 기대
  - 따라서, 실시간 채팅이 쌓일 때마다 최근 N개(window_size)의 문장으로 승률을 예측

Data Description

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Crawling
  - 2018년 2월부터 2020년 4월까지의 LCK Korea 영상 목록 + 해당 영상의 실시간 채팅
  - meta data : 각 영상별 경기수, 각 경기의 대진팀+선택한 캐릭터+선수 이름, 채팅자+채팅시간



Data Description

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

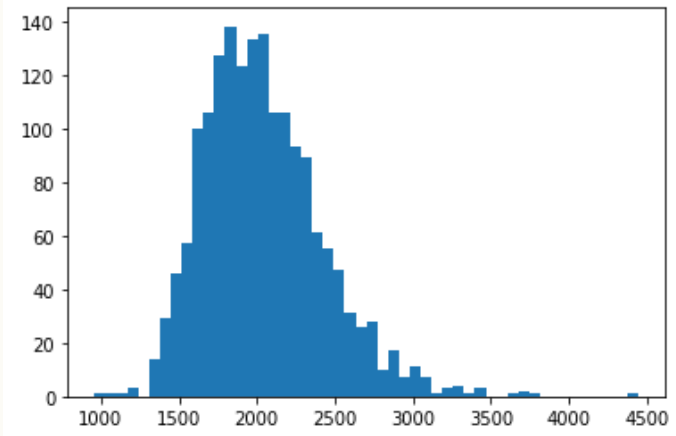
- Preprocessing
  - 팀명, 캐릭터명, 선수명이 들어간 단어를 한 단어로 치환
    - 캐릭터나 선수의 개인기량이 반영되기 때문에 삭제하고 채팅 내용만 반영
    - 치환 여부에 따라 실험하였으나 치환을 한 것의 결과가 더 좋음
  - 치환한 단어가 들어간 채팅만 선별

Data Description

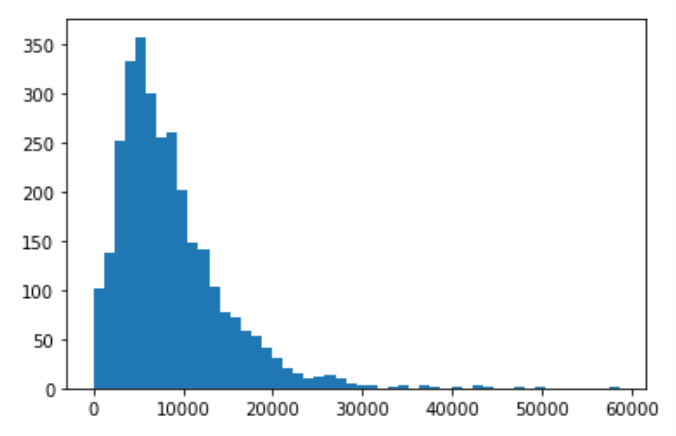
LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Document

영상 개수	340
경기 개수	3,036
경기 평균 채팅수	8,507



경기별 시간(초)
*평균 34분



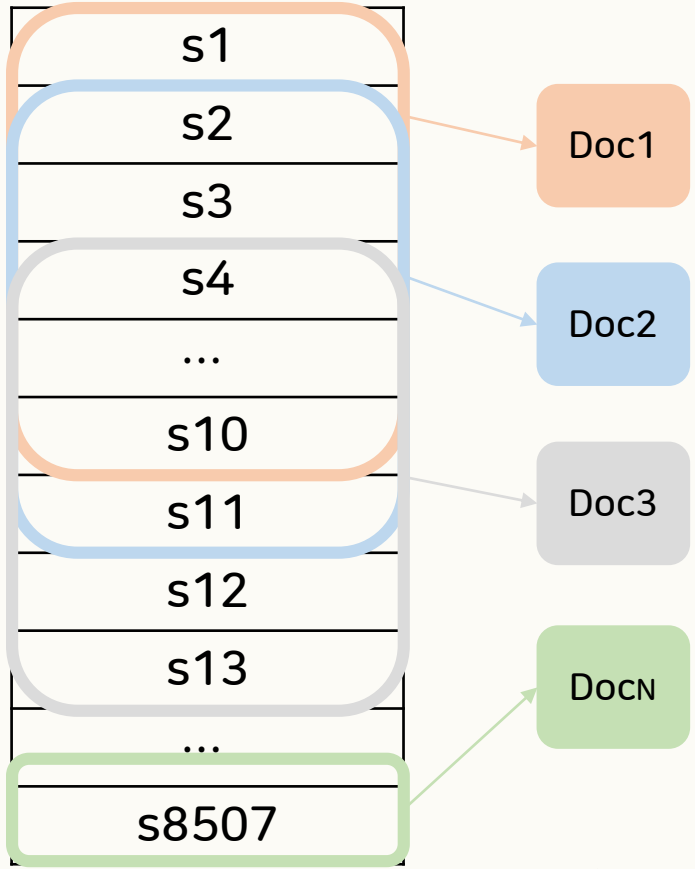
경기별 채팅수
*평균 8,507개

- 시간이 지남에 따라 승패를 예측하기 위해 최근 N개의 문장(window_size)로 나누어 Document 생성

Overall Architecture

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Design of Experiment



* window_size만큼 split

KoGPT2 Embedding
+
LSTM/GRU

Hierarchical Attention Network

BERT
(multilingual / KoBert SKT)

승/패 예측
(Acc 계산)

Overall Architecture

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

Crawling

- Twitch LCK Korea 채널 영상의 각 경기별 실시간 채팅 수집

Preprocessing

- 각 팀의 팀이름, 선수명, 캐릭터명이 들어간 채팅만 선별
- 승패팀의 문서를 분리

Embedding

- 모든 채팅을 단어 단위로 word2vec 생성
- 사용자 사전으로 같은 팀을 지칭하는 동의어 리스트와 롤에 사용되는 용어들을 추가
- pre-tained된 BERT / GPT embedding 사용

Modeling

- 한 경기의 채팅을 문장단위 시퀀스로 input 설정
- 시퀀스 모델을 사용하여 승, 패 분류

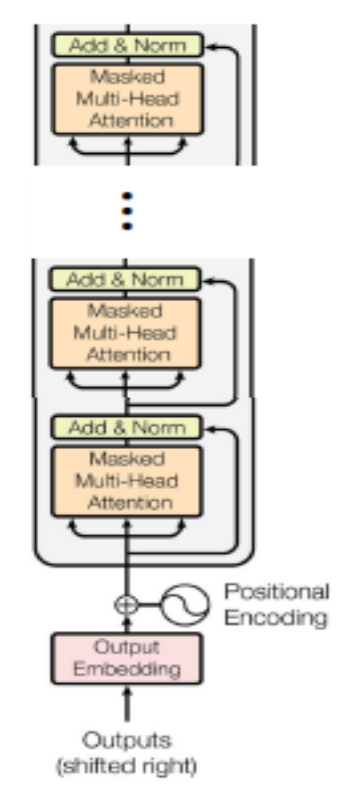
Evaluating

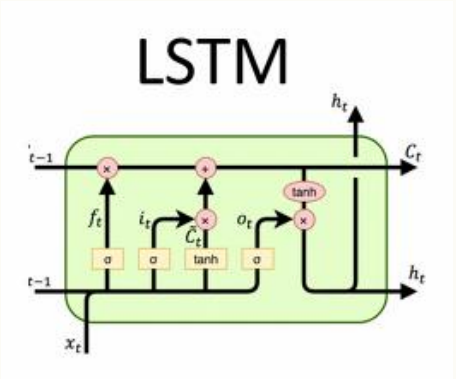
- 채팅을 입력하면 승, 패의 확률값이 출력
- 실제 승패와 비교하여 Acc 계산

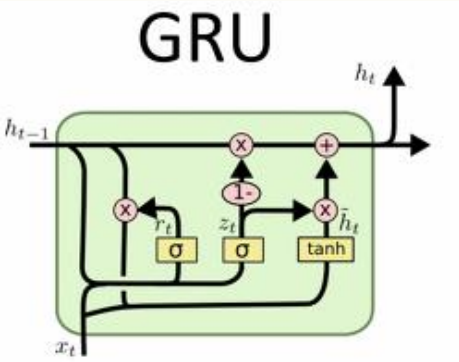
Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2
- LSTM / GRU







실시간 채팅 TEXT → KoGPT2 → Embedding vector → LSTM GRU → 분류

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + GRU
  - pre-trained된 embedding vector를 사용하여 승패 예측
  - epoch 100으로 통일

embedding size	layer1 dim	layer2 dim	dropout	train Acc	test Acc
100	64	16	0.1	0.8310	0.5620
100	64	16	0.6	0.7966	0.5532
300	64	16	0.1	0.8506	0.5592
300	64	16	0.6	0.7867	0.5593

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + Bi-GRU
  - pre-trained된 embedding vector를 사용하여 승패 예측
  - epoch 100으로 통일

embedding size	layer1 dim	layer2 dim	dropout	train Acc	test Acc
100	64	16	0.1	0.856	0.558
100	64	16	0.6	0.8588	0.5589
100	128	32	0.1	0.8865	0.5531
100	128	32	0.6	0.8339	0.5491
300	128	32	0.1	0.9139	0.5523
300	128	32	0.6	0.8761	0.5574

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + Bi-GRU + 치환X
  - pre-trained된 embedding vector를 사용하여 승패 예측
  - epoch 100으로 통일

embedding size	layer1 dim	layer2 dim	dropout	train Acc	test Acc
100	64	16	0.1	0.8685	0.5460
100	64	16	0.6	0.8512	0.5243
300	128	32	0.1	0.8993	0.5493
300	128	32	0.6	0.8012	0.5012

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + LSTM
  - pre-trained된 embedding vector를 사용하여 승패 예측
  - epoch 100으로 통일

embedding size	layer1 dim	layer2 dim	dropout	train Acc	test Acc
100	64	16	0.1	0.8491	0.5596
100	64	16	0.6	0.8184	0.5245
300	64	16	0.1	0.8685	0.5580
300	64	16	0.6	0.8514	0.5547

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + Bi-LSTM
  - pre-trained된 embedding vector를 사용하여 승패 예측
  - epoch 100으로 통일

embedding size	layer1 dim	layer2 dim	dropout	train Acc	test Acc
100	64	16	0.1	0.8565	0.5578
100	64	16	0.6	0.8312	0.5551
300	64	16	0.1	0.9060	0.5547
300	64	16	0.6	0.8326	0.5487
300	128	32	0.1	0.8939	0.5553
300	128	32	0.6	0.8729	0.5569

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- KoGPT2 + MODEL
  - **epoch 100**

model	train Acc	test Acc
KoGPT2 + GRU	0.8310	0.5620
KoGPT2 + Bi-GRU	0.8588	0.5589
KoGPT2 + Bi-GRU + 치환x	0.8993	0.5493
KoGPT2 + LSTM	0.8491	0.5596
KoGPT2 + Bi-LSTM	0.8565	0.5578

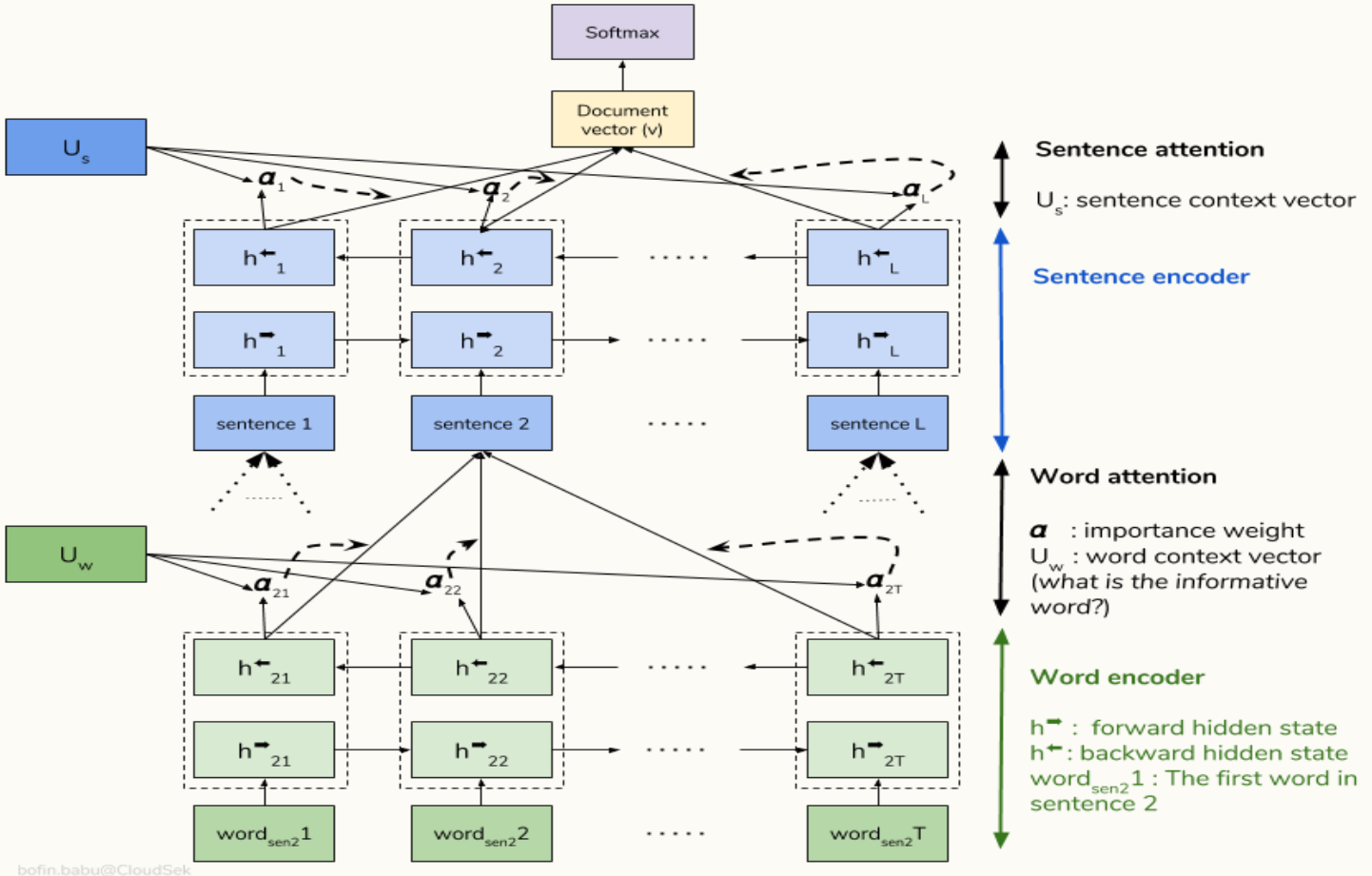
Result

1. 성능이 거의 차이가 없지만 GRU 모델의 성능이 가장 좋음
2. 치환 여부가 크게 영향을 미치지 않음
   → KoGPT2에서 치환된 단어가 실제 단어로 제대로 매핑되지 않기 때문
3. train과 test의 Acc 차이가 큼
   → 과적합 의심

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Hierarchical Attention Networks
  - Word2Vec 구축
    (256D / 512D)
  - LSTM / Bi-LSTM
  - L1-L2 Regularizations



Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- **Hierarchical Attention Networks (LSTM)**

epoch	word2vec 256D dropout 0.1 L1-L2 norm X		word2vec 256D dropout 0.6 L1-L2 norm X		word2vec 512D dropout 0.6 L1-L2 norm X		word2vec 512D dropout 0.6 L1-L2 norm (l1:1e-5, l2:1e-4)		word2vec 512D dropout 0.6 L1-L2 norm (l1:1e-3, l2:1e-3)	
	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc
10	0.685	0.592	0.653	*0.623*	0.704	*0.646*	0.686	*0.631*	0.692	*0.642*
20	0.701	0.506	0.670	0.620	0.723	0.638	0.726	0.626	0.713	0.641
50	0.728	*0.607*	0.725	0.602	0.789	0.610	0.782	0.597	0.761	0.608
100	0.796	0.520	0.786	0.594	0.801	0.557	0.858	0.580	0.497	0.474

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- **Hierarchical Attention Networks (Bi-LSTM)**

epoch	word2vec 256D dropout 0.1 L1-L2 norm X		word2vec 256D dropout 0.6 L1-L2 norm X		word2vec 512D dropout 0.6 L1-L2 norm X		word2vec 512D dropout 0.6 L1-L2 norm (l1:1e-5, l2:1e-4)		word2vec 512D dropout 0.6 L1-L2 norm (l1:1e-3, l2:1e-3)	
	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc	train Acc	test Acc
10	0.713	0.626	0.676	0.634	0.693	0.630	0.704	*0.636*	0.503	0.526
20	0.758	*0.661*	0.688	*0.638*	0.729	*0.632*	0.683	0.615	0.503	0.526
50	0.824	0.578	0.730	0.630	0.755	0.561	0.783	0.600	0.707	*0.610*
100	0.866	0.556	0.825	0.595	0.921	0.587	0.915	0.580	0.902	0.588

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Hierarchical Attention Networks
  - epoch 100

model		train Acc	test Acc
LSTM	word2vec 256D dropout 0.1 L1-L2 norm X	0.769	0.520
	word2vec 512D dropout 0.6 L1-L2 norm X	0.801	0.557
Bi-LSTM	word2vec 256D dropout 0.1 L1-L2 norm X	0.866	0.556
	word2vec 512D dropout 0.6 L1-L2 norm X	0.921	0.587

+	window_size 100 word2vec 512D dropout 0.6 L1-L2 norm X	
epoch	train Acc	test Acc
1	0.5927	0.5062
5	0.8433	0.4855
10	0.9486	0.5290
15	0.9612	0.5293

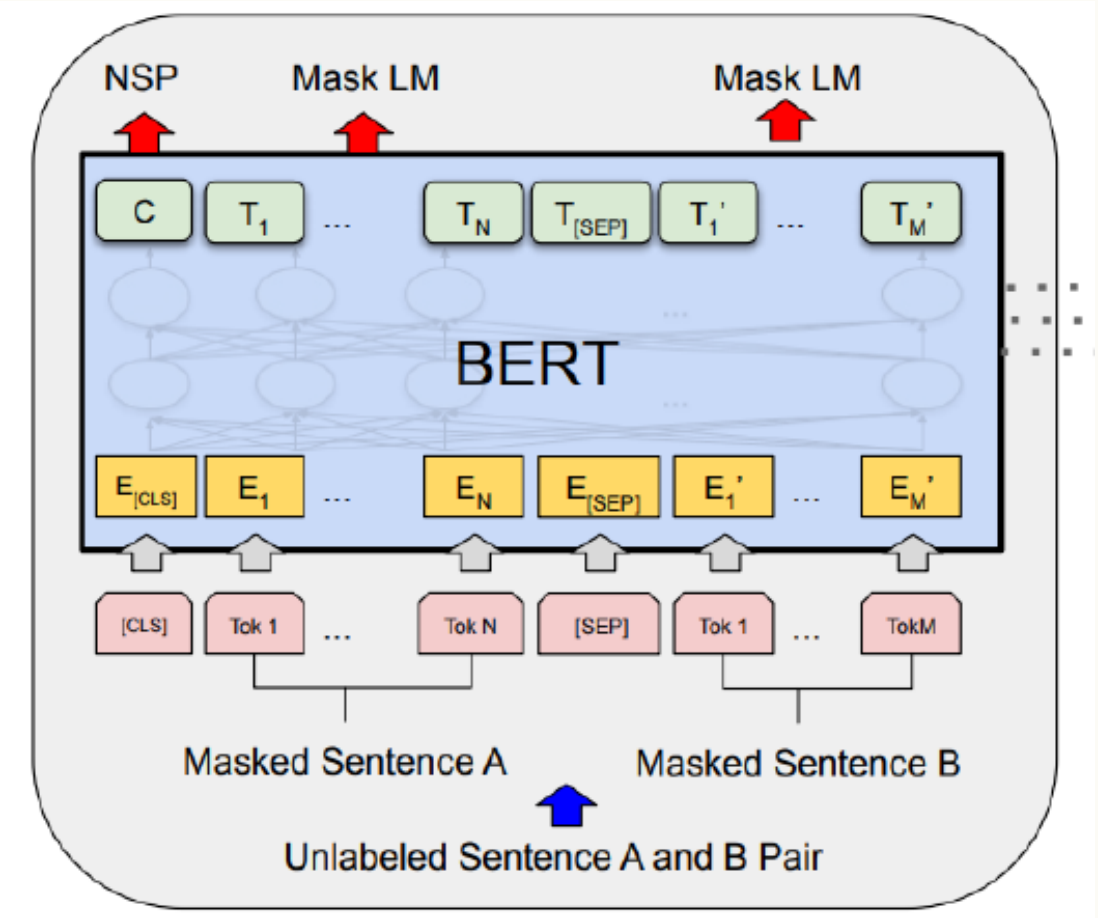
Result

1. 학습이 진행될수록 과적합이 보여 dropout 비율을 조절하고 L1-L2 regularization을 진행하였으나 큰차이가 없었음
2. word2vec 256D보다 512D의 성능이 더 좋음
3. window_size 100의 성능이 더 낮음

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- BERT (Base 12 layer)
  - Multilingual
  - KoBERT SKT ver.



Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- BERT

Multilingual BERT				
epoch	sampling 0.05		sampling 1	
	train Acc	test Acc	train Acc	test Acc
1	0.653	0.625	0.789	*0.643*
5	0.712	0.646	0.928	0.614
10	0.789	0.633	0.979	0.608
15	0.811	0.615	0.99	0.614

KoBERT SKT ver.				
epoch	dropout 0.1		dropout 0.6	
	train Acc	test Acc	train Acc	test Acc
1	0.599	0.675	0.579	0.669
2	0.676	0.697	0.671	0.692
3	0.703	0.734	0.698	0.728
4	0.736	0.784	0.715	*0.742*

Experiment Result

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- BERT
  - epoch 5(multilingual) & 4(KoBERT SKT)

model		train Acc	test Acc
Multilingual BERT	sampling 0.05	0.712	0.646
	sampling 1	0.928	0.614
KoBERT SKT ver.	dropout 0.1	0.736	0.784
	dropout 0.6	0.715	0.742

Result

1. 실시간 채팅이 한국어여서 Multilingual보다 KoBERT가 성능이 좋음
2. Multilingual에서 sampling을 하더라도 성능이 좋음
3. epoch 4에 Acc가 78%까지 높아짐

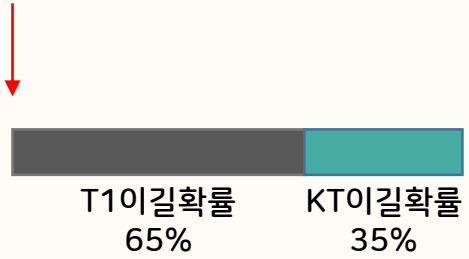
Conclusion

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- 결과 비교

T1 vs. Kt Rolster 전적			
경기수	승	패	승률
80	52	28	65%

롤 인벤에서 상대 전적 크롤링



승률을 기반으로 랜덤하게 승패 생성

predict	actual
0	0
0	1
1	1
0	0
0	0

실제값과 비교하여 Acc 계산

	Acc
Probabilitstic	0.569

Conclusion

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- **결과 비교**

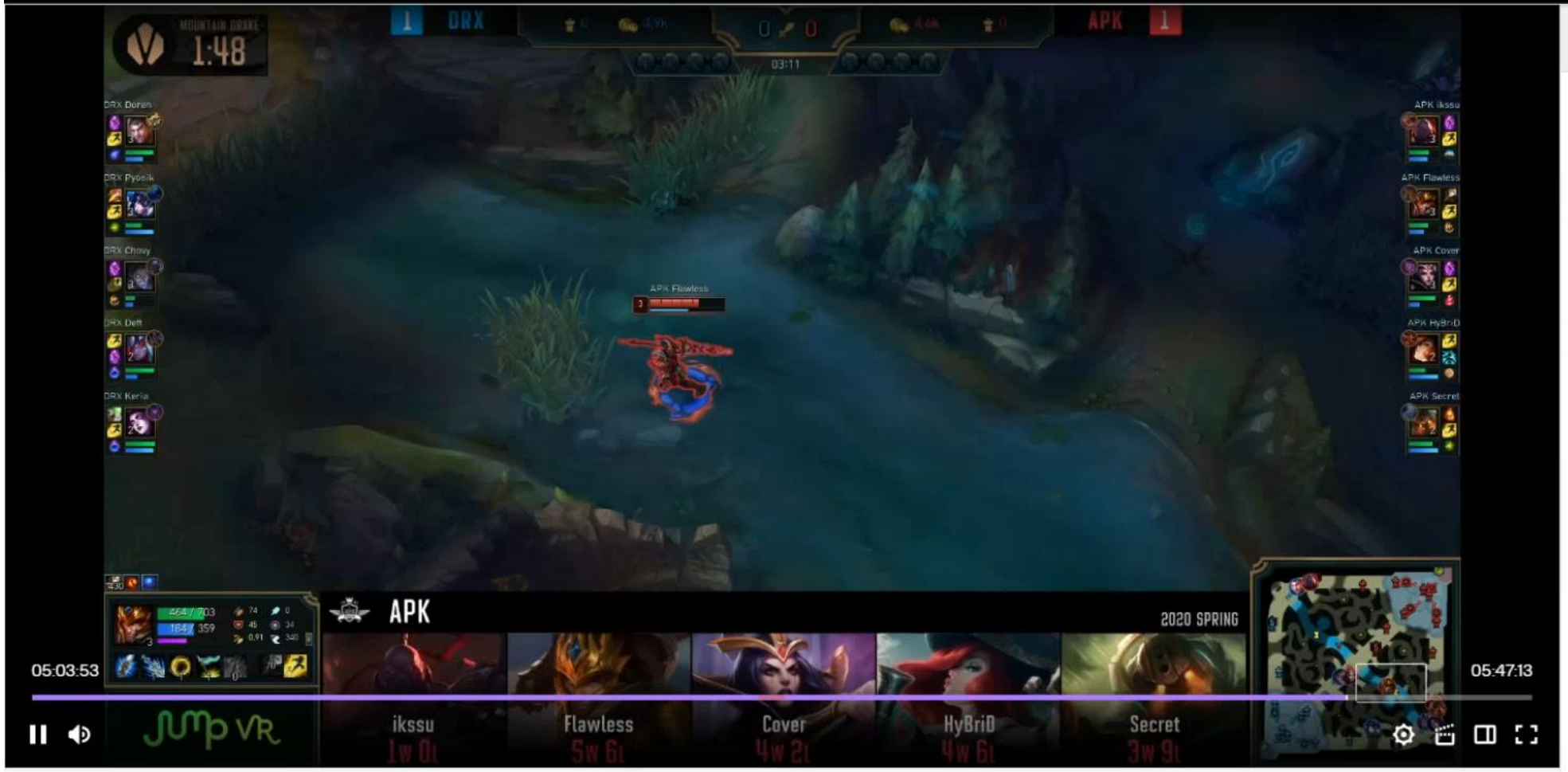
model	train Acc	test Acc
Probabilitstic		0.569
KoGPT2 + GRU	0.8310	0.5620
KoGPT2 + Bi-GRU	0.8588	0.5589
KoGPT2 + LSTM	0.8491	0.5596
KoGPT2 + Bi-LSTM	0.8565	0.5578
Hierarchical Attention Networks (LSTM)	0.704	0.646
Hierarchical Attention Networks (Bi-LSTM)	0.758	0.661
Multilingual BERT	0.712	0.646
KoBERT SKT ver.	0.736	0.784

Conclusion

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- Result
  - 대진팀의 전적을 이용하여 승패를 예측하는 것보다 실시간 채팅을 이용하여 예측하는 것이 좀 더 정확함
  - KoGPT를 embedding으로 사용하는 것은 train set과 test set 사이의 Acc 차이가 큼
  - LSTM만 비교하였을 때, Hierarchical Attention이 문장과 단어의 중요성을 모두 고려하여 성능이 더 좋음
  - KoBERT의 성능이 가장 좋음

- Limitation
  - BERT의 학습을 더 많이 시켰다면 성능이 높아졌을 것이라고 기대
  - 과적합을 해결하기 위한 추가 실험이 필요
  - window_size를 더 다양하게 실험하면 적합한 채팅의 길이를 알 수 있을 것



[2020 우리은행 LCK Spring Split] GRF vs. HLE - DRX vs. APK · 2개월 전

League of Legends

52,470 공유

LCK_Korea

동영상 채팅

5:03:31 작은운하 (littlecanal): 님이 해서 그런거아님

5:03:33 닭똥집튀김 (kwt0323): 르블랑 스킨뭔가요???

5:03:33 프라이데이지 (arcadiam): ㅋㅋㅋㅋㅋㅋㅋㅋㅋ 사붕이 두들겨맞는다

5:03:34 예쁜말만하기 (hanadul3net): QE야 미안한데

5:03:35 피즈는이성진 (dltjdwls5507): 피즈즐지

5:03:36 안녕빌런 (cha00629): 픽들 승률 미쳤네

5:03:37 전범맨 (wind3592): 방관 바루스

5:03:41 치킨맛간장맛치킨 (wldnd76): 와

5:03:42 binbang18: 81

5:03:42 WHiteCosmos1: 전겜은 몰라도 이겜은 지면 안되는디

5:03:43 차라투스트라는이렇게말했 (gjsqjq): 피즈 3티어

5:03:43 레이시크 (rhqudtn246): 무관 kt 또는 담원에 져 4위확정 ㅋㅋㅋㅋㅋㅋㅋㅋㅋ

5:03:46 초코맛바나나우유 (ssang574): 정글차이 ㅋㅋㅋ 이 팀은 무슨 돌아가면서 경험치를 먹냐 ㅋㅋ

5:03:47 감성센도 (kdh3710): 와 물가 저 스킨 뭐야 진짜 얘쁘네

5:03:48 고니구니 (itn1027): 거의 베스트 픽 ㄷ

5:03:49 khs224: 와드한척ㅋㅋㅋ

5:03:51 집떠나와열차타고 (dldudwlsz): 메카너프되서

5:03:52 prozypark: 이번에 표식 제대로 배우겠네

5:03:53 reverie47: 지금 apk젠도 t12위임??

Conclusion

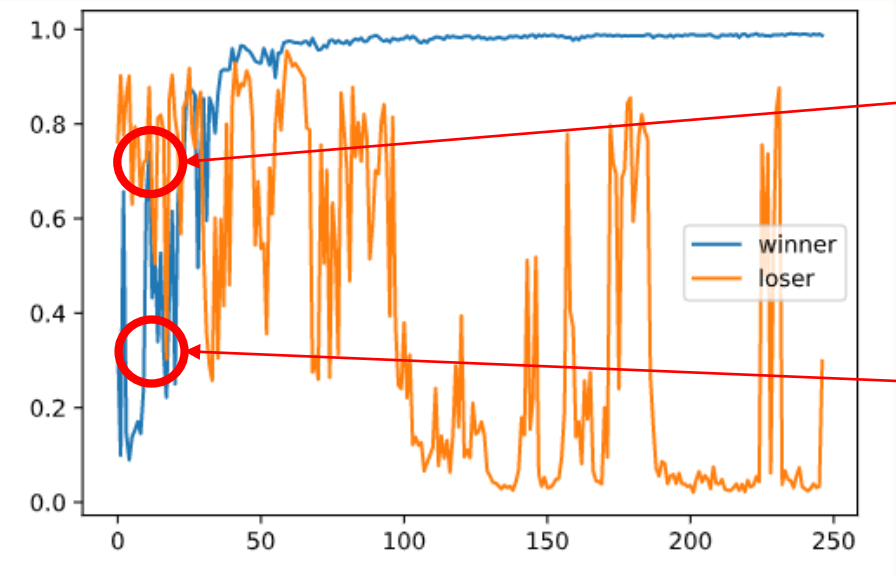
LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- DEMO – KoBERT SKT ver. epoch 4

**48초 소요

  - 채팅이 쌓일 때마다 마지막 10개의 채팅을 input으로 사용하여 그 때의 이길 확률을 계산

* 각 팀을 나타내는 치환 단어가 포함된 채팅을 사용하였기 때문에 이긴 팀과 진 팀의 Document는 다름



이긴 팀의 이길 확률이 0.4에서 0.7~8로 급상승
진 팀의 공격이 실패하면서 위치를 상대에게 들킴
→ 이긴 팀에게 유리한 작용

진 팀의 이길 확률이 0.6에서 0.2로 급하락
진 팀의 공격이 실패하면서 진 팀을 조롱, 무시하는 채팅 등장
ex, 표식 하는게뭐임?
표식이 3세_ 내내 계속 밀리긴 하네

Conclusion

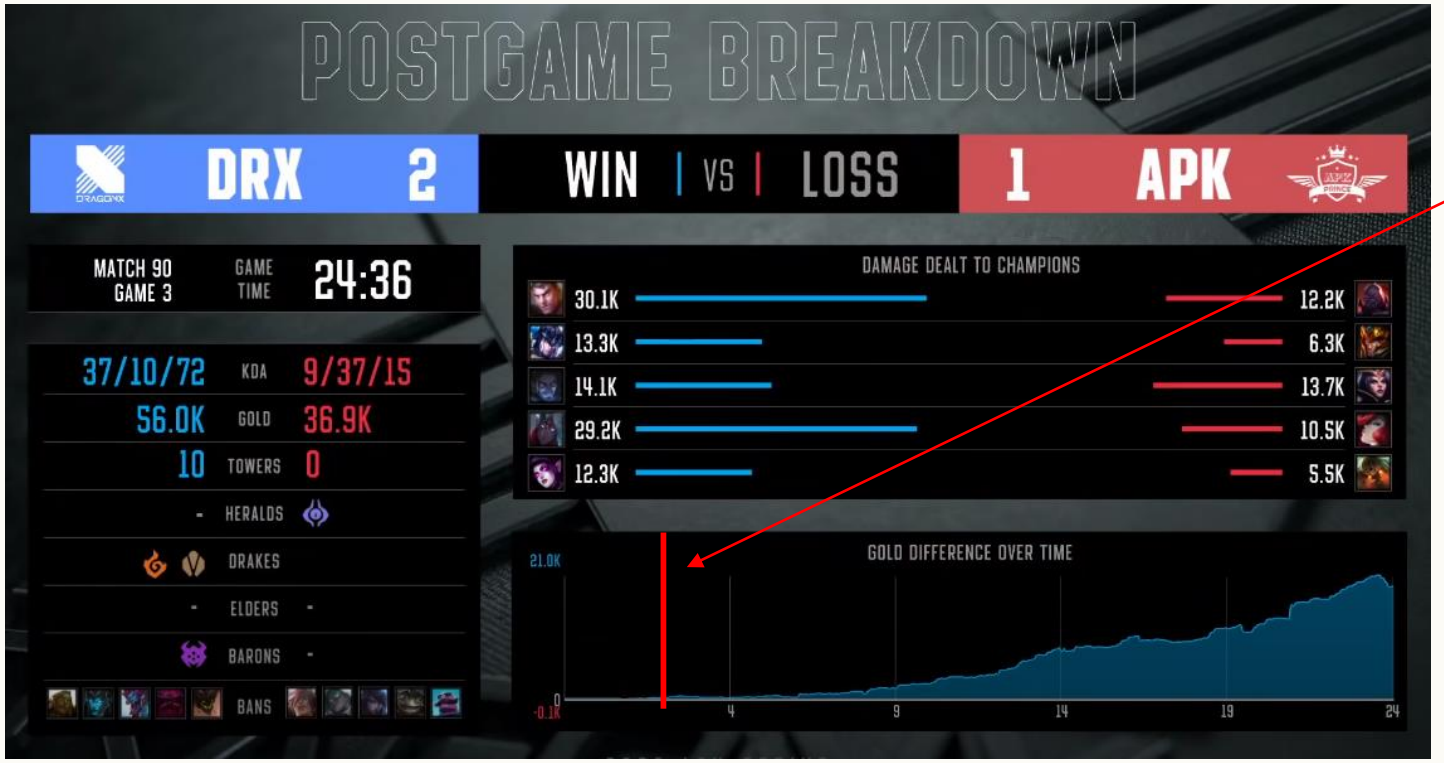
LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- DEMO – KoBERT SKT ver. epoch 4

**48초 소요

  - 채팅이 쌓일 때마다 마지막 10개의 채팅을 input으로 사용하여 그 때의 이길 확률을 계산

* 각 팀을 나타내는 치환 단어가 포함된 채팅을 사용하였기 때문에 이긴 팀과 진 팀의 Document는 다름



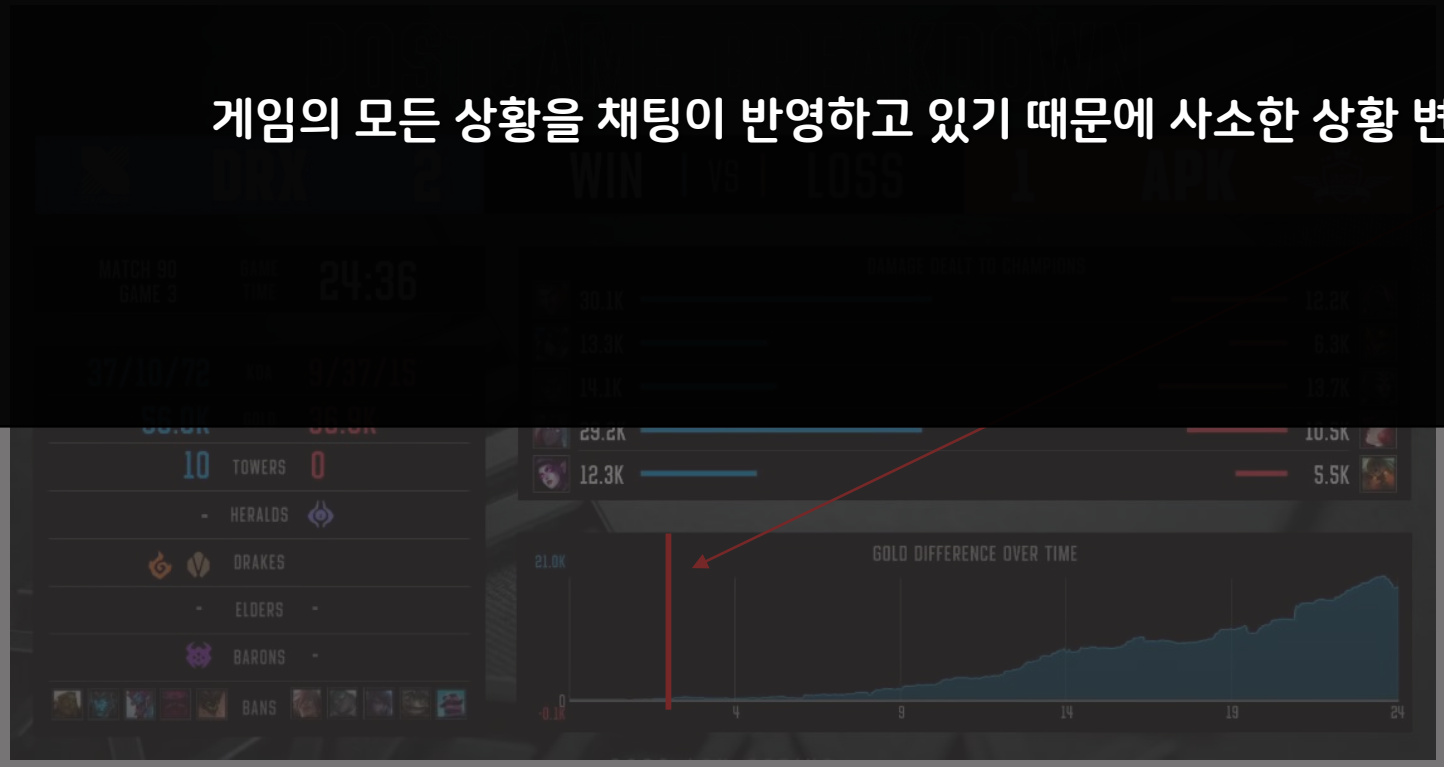
상황이 발생했던 3~4분에 메타데이터는 아무런 변화가 없었지만,
KoBERT에서는 이길 확률에 대한 변화를 캐치하였음.

Conclusion

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- DEMO – KoBERT SKT ver. epoch 4 **48초 소요
  - 채팅이 쌓일 때마다 마지막 10개의 채팅을 input으로 사용하여 그 때의 이길 확률을 계산
  - *각 팀을 나타내는 치환 단어가 포함된 채팅을 사용하였기 때문에 이긴 팀과 진 팀의 Document는 다름

게임의 모든 상황을 채팅이 반영하고 있기 때문에 사소한 상황 변화에도 민감하게 반응함



피드백

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- PPT에 반영된 것
  - “LSTM 모델을 input, output을 어떻게 주어 sequence의 승률 변화를 낼지 구체적인 제시가 안되어 있어 아쉬웠습니다.”
    - 답변 : 7페이지를 참고해주세요.
  - “실시간으로 승률을 예측한다고 하셨는데 실시간 승률 예측의 정확도는 어떻게 평가할지 궁금합니다.”
    - 답변 : 24, 27페이지를 참고해주세요. 평가는 강필성 교수님께서 조언해 주신 내용 + 실제 데모 영상을 반영하였습니다.
  - “어떤 채팅이 승패에 관련된 흐름에 관한 것인지 프로게이머의 플레이에 대해 관련된 것인지 어떻게 구별할 수 있나요? 예를 들어 A팀은 지고 있는데 A팀의 한 선수만이 뛰어난 플레이를 해서 칭찬을 많이 받아 이것이 스코어가 많이 들어가 승패를 잘 못 예측 할 수도 있다고 생각합니다. “
    - 답변 : 5페이지를 참고해주세요. 팀별로 모두 같은 단어로 치환하였습니다.
  - “발표해주신 내용으로 이해하자면 모든 댓글들은 순서를 가지고 있고 전처리를 통해 팀 이름, 캐릭터명, 선수 이름이 언급된 채팅이 존재한 것만 남겨주신다고 하였습니다. 인터넷 채팅은 워낙 빠르게 쓰여지고 있어 사용자들이 동시에 댓글을 입력하는 상황도 있을 수 있습니다. 이러한 경우가 sequence를 가지고 있는 것이라 말할 수 있는지 의문이 듭니다.”
    - 답변 : 7페이지를 참고해주세요. window_size로 묶어서 계산하여, 동시에 댓글을 입력하는 상황도 고려하였습니다.

피드백

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- **PPT에 반영된 것**
  - “피드백 부분에서 동의어 리스트를 직접 만들어 준다고 하였는데, 이 부분은 직접 수작업으로 만든다는 내용인지 궁금합니다. 수작업으로 만든다면 동의어가 상당히 많이 나올텐데 이를 어떻게 처리할 지 궁금합니다. 프로젝트 발표 Architecture부분에서 팀 관련된 모든 대명사를 하나로 통일해준다고 하였는데, 이럴 경우에는 굳이 동의어 사전을 만들 필요가 없지 않을까 생각됩니다.”
    - 답변 : 동의어에 경우 영상 및 채팅을 하나하나 읽으며 찾았습니다. 이렇게 찾은 동의어 들을 Document에서 직접적으로 바꾸었습니다.
      [ex) 슥 -> T1] 동의어 사전을 만든 이유는 모든 대명사를 하나로 통일해주기 위해서 입니다. [ex) 슥 -> T1 -> 이용택, ex2) 슥 -> 슥]
      보여드리는 예시와 같이 동의어 사전을 만들지 않았다면 슥은 이용택이 될 수 없습니다. 이용택이 된 이유는 5페이지를 참고해주세요.
  - “현재 모델의 평가는 마지막 승패만으로 판단되는 것처럼 보이는데, 시간단위별 승패 진행상황 평가는 정답데이터를 어떻게 구축하여 평가할 것인지 궁금합니다”
    - 답변 : 7페이지를 참고해주세요. 댓글이 만들어지는 상황별로 진행상황 평가를 진행하였습니다.
  - “사용자사전에 사용자들이 채팅에 자주 사용하는 단어들을 추가했다고 하였는데 총 1250만개의 문장이 존재하기 때문에 사용자사전에 추가하는 것 외에 추가적인 전처리 작업이 필요할 것 같습니다.”
    - 답변 : 동의어 사전 구축 등 전처리 작업을 진행하였습니다.
  - “실시간으로 승패를 예측한다는 말과 실시간 중계와 다른 점이 궁금합니다. “
    - 답변 : 28페이지를 참고해주세요. 실시간 중계에서는 경기에 큰 영향을 주는 사건 위주로 진행되는데, 저희 모델은 그렇지 않은 것을 확인하였습니다.

피드백

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- PPT에 반영된 것
  - “실시간 채팅만을 이용하였을 경우, 다른 정형데이터를 사용하였을 때에 비해 성능이 얼마나 나왔는지에 대한 것”
    - 답변 : 정형데이터를 사용하였을 때의 데이터를 사용하여, 성능을 비교하였습니다.
  - “실시간 채팅이 바로 모델에 반영되고 모델의 학습속도가 그만큼 빠른지 궁금합니다.(실시간으로 하려면 그때마다 워드 임베딩을 어떻게 할 건지에 대해서 궁금합니다.)”
    - 답변 : 워드 임베딩은 미리 해두었기 때문에 따로 할 필요 없었으며, KoBERT기준으로 한 경기를 예측하는 데에 48초가 소요되었습니다. 한 문서는 약 2600개였습니다. 따라서 1개의 채팅을 예측할 때에는 1초 미만이 소요될 것으로 예상됩니다.

피드백

LoL 방송의 실시간 채팅을 이용하여 승패를 예측하라!

- PPT에 반영되지 않은 것
  - “overall architecture에서 document의 label이 승패에 해당하는 label과 어떠한 연관성이 있는지 질문 드립니다.”
    - 답변 : 승에 해당하는 document는 라벨을 1로 패에 해당하는 document는 라벨을 0으로 하였습니다.
  - “N-gram을 이용해 보는 것은 어떨까요? 토큰 하나의 의미보다는 토큰 두개를 하나의 단어로 보고 Embedding 했을 때 더 명확한 의미를 Embedding할 수 있습니다. 예를 들어 "대박"은 좋은 의미, "노잼"은 나쁜 의미라고 했을 때 "대박 노잼" 아주 나쁜 의미로 사용됩니다. 따라서 N-Gram을 이용하는게 더 높은 모델을 만들수 있을것 같아 보입니다.”
    - 답변 : 기존에 준비했던 모델을 구현하고 학습하는 데에 오랜 시간이 걸려 피드백을 반영하지 못했습니다.
  - “첫째, 검증데이터의 편향성을 방지하기 위해 학습데이터와 검증데이터의 비율을 8:2로 고정하지 않고 cross-validation 적용을 추천드립니다. 둘째, 정형데이터를 사용하는 것도 고려해보셨으면 됩니다. 정형데이터를 사용한 모델보다 정형 + 비정형 데이터를 함께 사용하는 모델이 더 낫다는 것을 보여주면 비정형데이터 활용 효과를 나타낼 수 있지 않을까요.”
    - 답변 : 시간을 고려하기 위해서 cross-validation은 적용하지 않았습니다. 두번째로 정형데이터는 일부러 모델에 사용하지 않았습니다. 정형데이터는 이 모델을 평가하는 데에 사용하였습니다.
  - “기존 승률이 좋은 게이머와 승률이 좋지 않은 게이머에 따라 모델의 성능이 차이가 날 수 있을 것이라고 생각했습니다. 추가적으로 분석이 가능하시다면 게이머에 따라 분류 성능의 차이가 있을지 궁금합니다. “
    - 답변 : 게이머별로 성능의 차이가 나지 않도록 하기 위해서, 치환을 하였습니다.
  - “해설자의 음성데이터를 이용하여 고려하는 아이디어도 좋은 것 같다고 생각이 듭니다. “
    - 답변 : 프로젝트 초반부에 고려하였었으나, 현재 모델의 성능이 나쁘지 않았기 때문에 진행하지 않았습니다.

E O D