Lexically Constrained Decoding

decoding 수행 시 어휘적 제약을 주어 NMT의 sequence generation 성능을 높임.

발표자: 이세영

2023. 08. 12

Contents

Grid Beam Search 이란?

Grid Beam Search란?

- **Grid Beam Search:** 어휘적 제약(lexical constraints)를 포함하는 beam search의 확장 알고리즘
- seqence 생성 모델이면 모두 사용 가능. (log probabilities를 최대화하여 토큰을 생성하는 방식이면 모두 가능!)
- **Lexical Constraints**: output sequence에 반드시 포함되어야만하는 **구(phrase)** 혹은 **단어(words)**이다.
- 이 방법은 모델 파라미터나 train data의 수정 없이 매우 간편하게 사용할 수 있는 방법임. (학습된 모델에 사용 가능)
- 다양한 실험을 통해 Neural Translation과 다양한 도메인에서 실현 가능하고, 유연하게 동작함을 증명함.

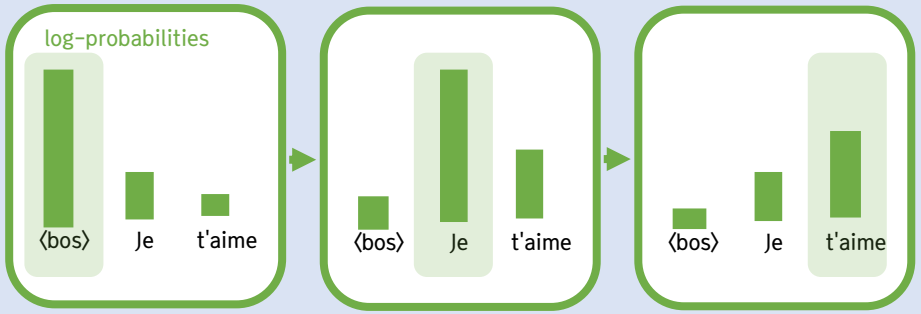
1. 제안배경

- 일부 실제상황에서는, inference time에 Optimal output sequence를 위한 추가적인 information을 추가할 수 있음. (Post-Editing)
  - 특정 domain input에서 해당 domain의 전문 용어가 output에 포함되었나 등 확인
- our goal: **lexical constraints 포함하면서도 좋은 퀄리티의 sequence 생성**
- MT usecase의 경우, **사용자의 입력**과 **모델의 output을 결합**하여 생성하는 경우가 많다.
- 본 논문에서는 Lexical Constraints의 개념을 공식(formalize)화하고, 모델의 output에 존재해야 하는 하위 시퀀스를 지정할 수 있는 decoding algorithm을 제안.
- 개별 제약은 단일 토큰 또는 다중 단어 구문(phrase)일 수 있으며, n개의 constraints을 동시에 지정할 수 있습니다.

2. 관련 연구

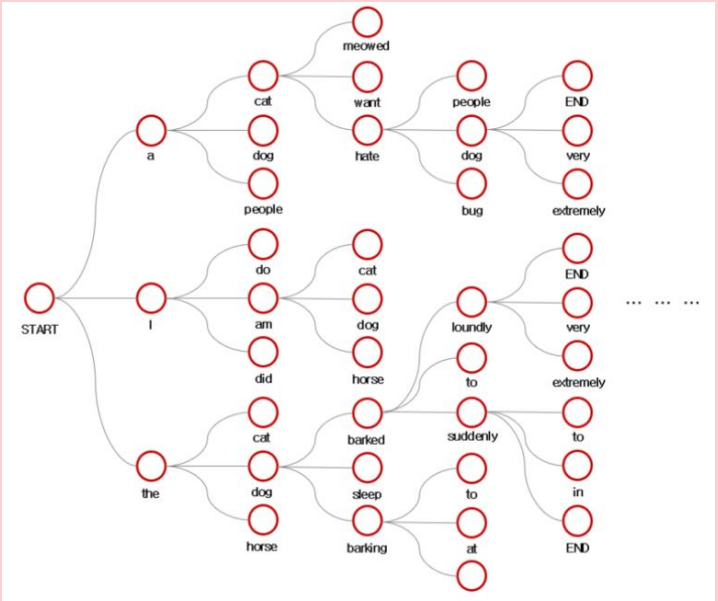
Greedy search

- 후보군 중에서 확률이 가장 높은 1개의 token 으로만 next token 선택

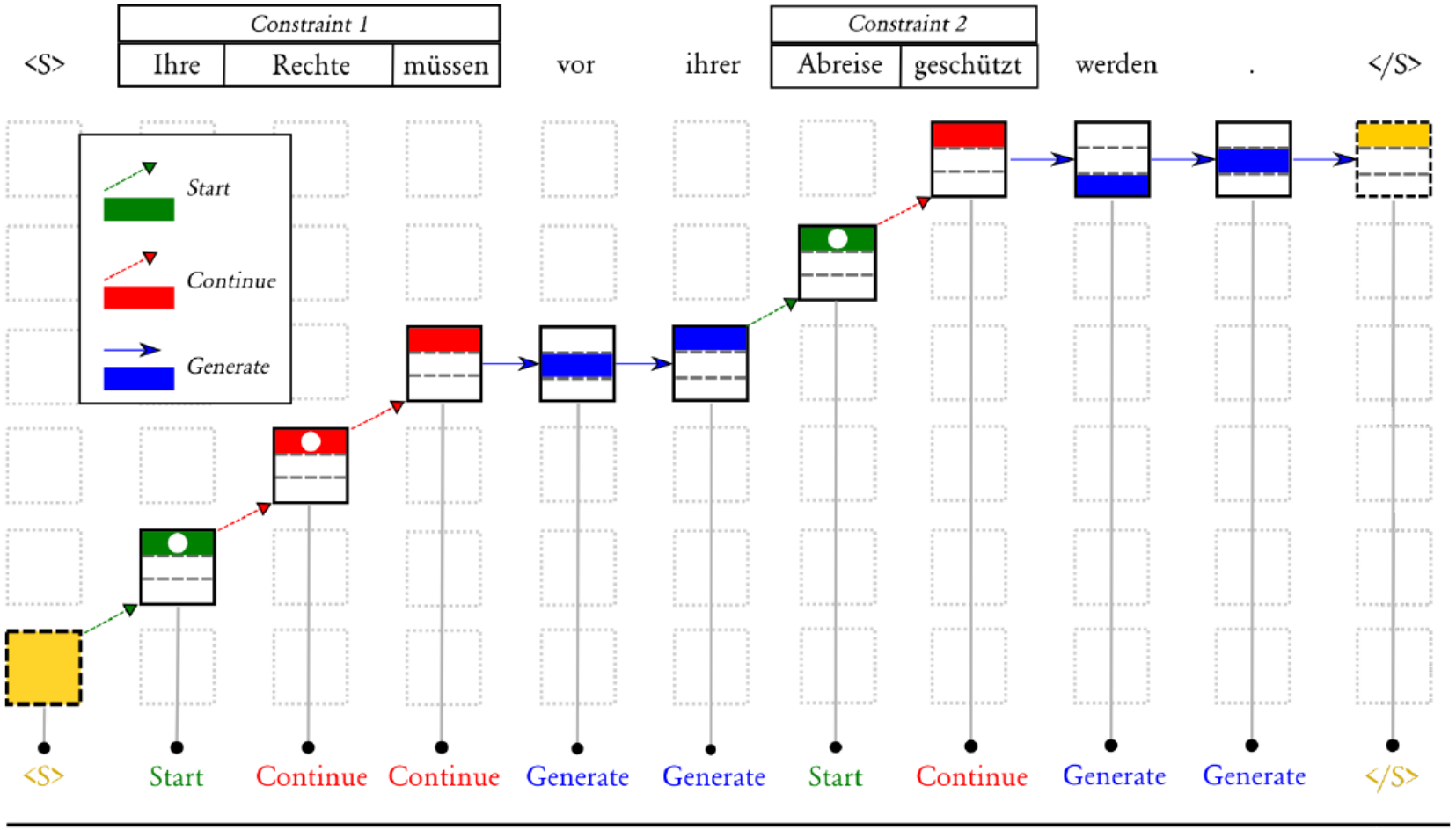


Beam Search

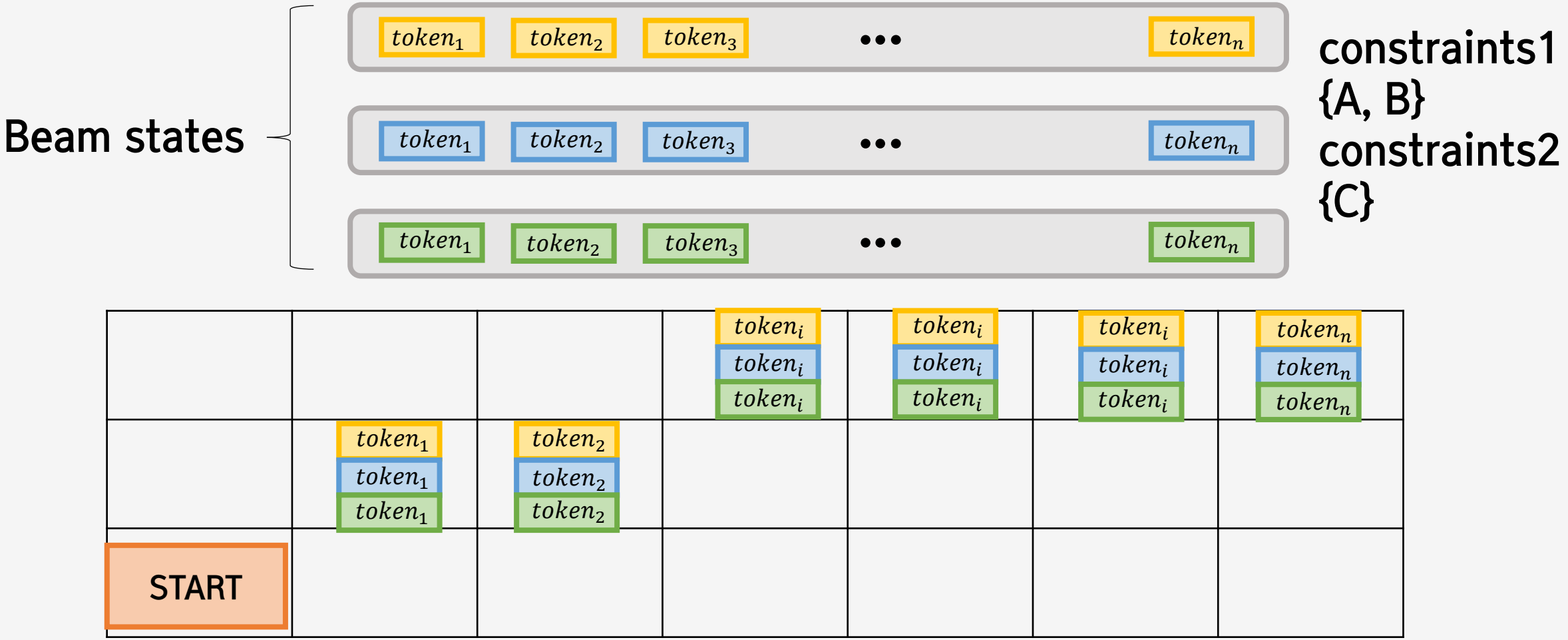
- n 개의 beam을 생성하여 search 수행



3. Grid Beam Search

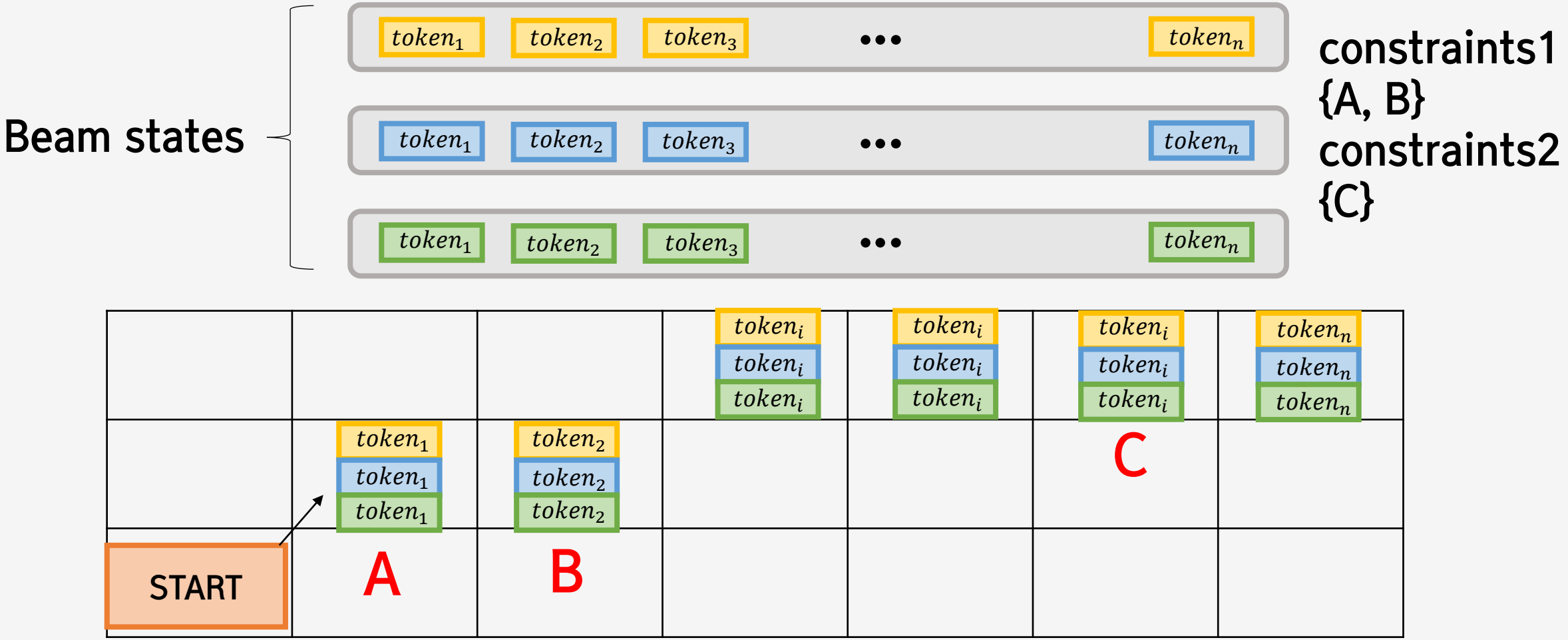


3. Grid Beam Search



Grid Beam Search(max_len=7, constraints=3)

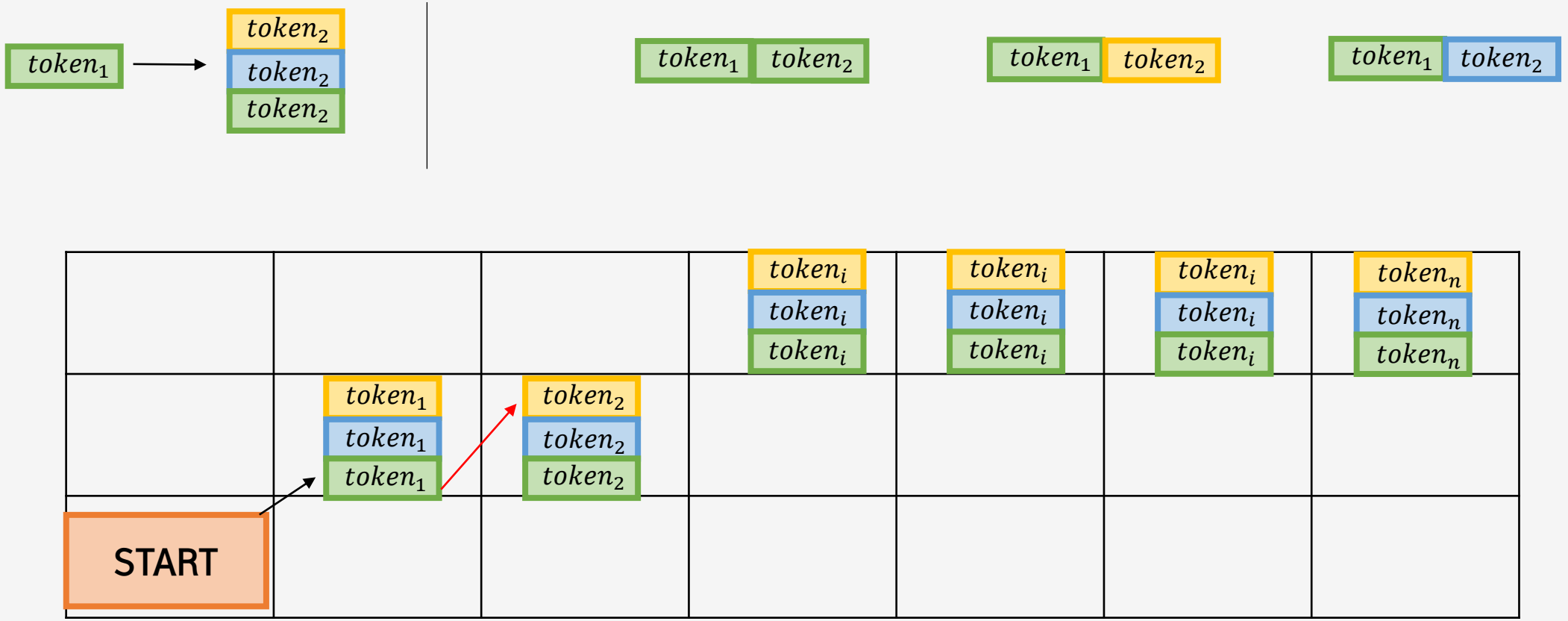
3. Grid Beam Search



Grid Beam Search(max_len=7, constraints=2)

3. Grid Beam Search

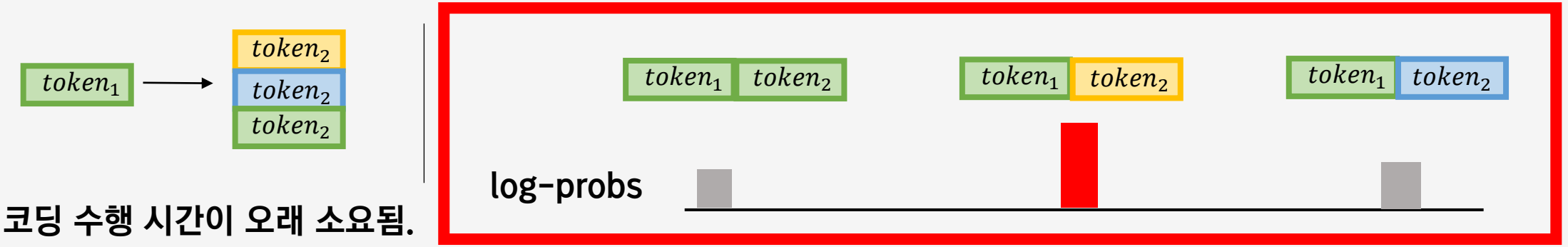
Beam을 고르는 기준: 현재까지 생성된 token+ new beam의 probabilities가 높은 것



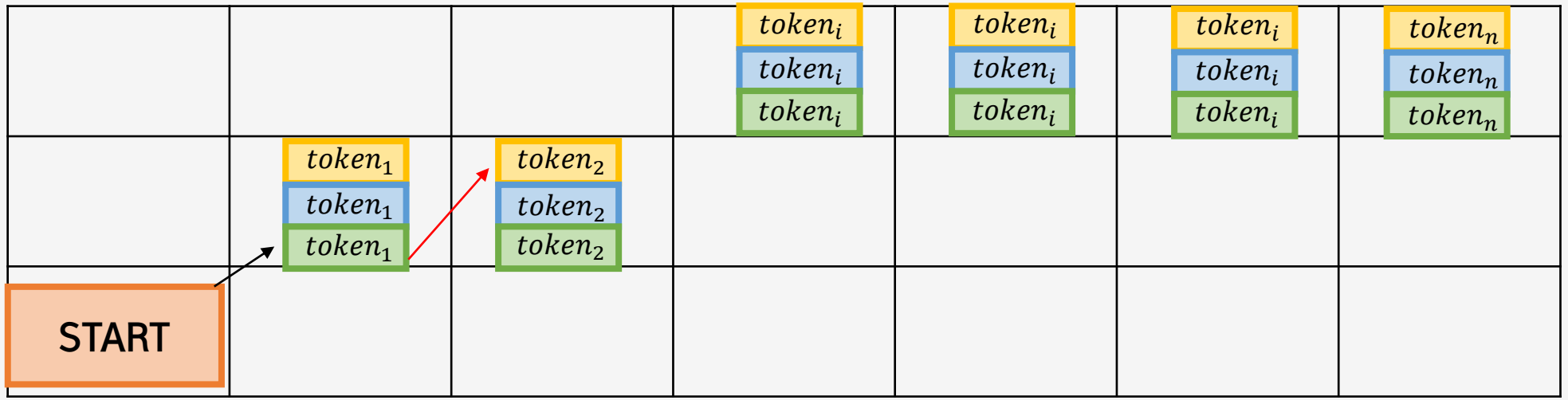
Grid Beam Search(max_len=7, constraints=2)

3. Grid Beam Search

Beam을 고르는 기준: 현재까지 생성된 token+ new beam의 probabilities가 높은 것



디코딩 수행 시간이 오래 소요됨.



Grid Beam Search(max_len=7, constraints=2)

3. Grid Beam Search

grid beam search implementation:
https://github.com/chrishokamp/constrained_decoding/blob/master/constrained_decoding/__init__.py#L200

4. Experiments – (1) Pick-Revise for Interactive Post Editing

Pick-Revise for Interactive Post Editing

- 번역 수행
- -> 사용자가 어색한 부분 수정
- -> 사용자가 제공한 어휘를 constraints로 사용
- -> constraints를 포함하여 번역문 생성

Source: May be it's a failure

Translation: 어쩌면 그것은 불합격.

사용자 수정: 어쩌면 실패일 것이다.

Constraints: ['실패일', '것이다.']

GBS Translation: 아마도 실패일 것이다.

ITERATION	0	1	2	3
Strict Constraints				
EN-DE	18.44	27.64 (+9.20)	36.66 (+9.01)	**43.92** (+7.26)
EN-FR	28.07	36.71 (+8.64)	44.84 (+8.13)	**45.48** +(0.63)
EN-PT*	15.41	23.54 (+8.25)	31.14 (+7.60)	**35.89** (+4.75)
Relaxed Constraints				
EN-DE	18.44	26.43 (+7.98)	34.48 (+8.04)	**41.82** (+7.34)
EN-FR	28.07	33.8 (+5.72)	40.33 (+6.53)	**47.0** (+6.67)
EN-PT*	15.41	23.22 (+7.80)	33.82 (+10.6)	**40.75** (+6.93)

Table: 4번 반복하였을 때의 BLEU score 비교

4. Experiments – (2) Domain Adaptation via Terminology

기존의 방법

- Case of a Male Newborn with Incontinentia Pigmenti Initially Misdiagnosed as a Recurrent Skin Infection

=> Case of a Male Newborn with <TERM_1> Initially Misdiagnosed as a Recurrent Skin Infection

전문용어를 NER처럼 다뤄 새로운 토큰으로 치환

-> 모델이 토큰을 학습할 수 있는 가능성을 잃음.

대안

- constraints를 지정하여 번역
- Random: Translation의 랜덤 위치에 용어 삽입
- Beginning: origin_target_sentence 의 시작 위치를 constraint 로 부여
- GBS: src-tgt 용어를 constraints로 부여

System	BLEU
EN-DE	
Baseline	26.17
Random	25.18 (-0.99)
Beginning	26.44 (+0.26)
GBS	**27.99** (+1.82)
EN-FR	
Baseline	32.45
Random	31.48 (-0.97)
Beginning	34.51 (+2.05)
GBS	**35.05** (+2.59)
EN-PT	
Baseline	15.41
Random	18.26 (+2.85)
Beginning	20.43 (+5.02)
GBS	**29.15** (+13.73)

5. Conclusion

- Constraints을 추가한 decoder의 output sequence에서 constraints 를 올바르게 배치 가능
- Lexically Constrained Decoding은 토큰 단위로 출력 시퀀스를 생성하는 모든 모델의 출력에 적용할 수 있는 flexible한 방법
- 번역기가 기존 hypos에 대한 수정을 제공할 수 있는 번역 인터페이스에서 사용자 입력은 constraints로 사용되어 사용자가 오류를 수정할 때마다 새로운 출력을 생성할 수 있다.
- 실험을 통해 번역 품질이 크게 향상되는 것을 증명함.
- 도메인별 용어를 constraints로 생성함으로써 일반 모델이 finetuning없이 새로운 도메인에 적응할 수 있음.
- 향후 작업에서는 자동 요약, 이미지 캡션 또는 대화생성과 같은 MT 외부의 모델로 GBS의 성능이 평가되기를 희망

Thank You