Търсене и извличане на информация. Приложение на дълбоко машинно обучение

Стоян Михов

Лекция 13: Условен езиков модел. Модел “Последователност към последователност” (Sequence to sequence). Архитектура “Внимание”

План на лекцията

1. **Формалности за курса (5 мин)**
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Формалности

- Това е предпоследната лекция на курса за този семестър
- Оценките и решенията на домашно задание №2 са публикувани в Мудъл
- Условието за домашно задание №3 ще бъде публикувано в Мудъл до края на седмицата.
- Очаквам решенията на домашно задание №3 да бъдат предадени до края на деня на 18.01.2023 г.
- Формалното условие за курсовата работа ще бъде публикувано следващата седмица.
- Лекция 13 се базира на глава 17 от втория учебник.

План на лекцията

1. Формалности за курса (5 мин)
2. **Обучение на условен модел (10 мин)**
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Условни модели

- Вероятностен модел (например езиков модел) е разпределение $\mathbf{Pr}[\mathbf{w}]$ върху елементарните събития $\mathbf{w} \in V^*$.

- Но в много задачи се налага за даден вход да се изведе съответен изход. Например за даден документ да се изведе съответен клас, за даден текст на български да се изведе съответен превод на английски, за дадена снимка да се изведе описание, за даден текст да се изведе резюме, за даден запис на реч да се изведе транскрипция и т.н.

- За тези приложения е необходимо да се моделира условно разпределение $\mathrm{Pr}[Y \mid X]$ — при даден вход X каква е вероятността за съответен извод Y.

- **Как да обучим условен модел?**

Обучение на условен модел

- Вероятностен модел обучаваме като минимизираме емпиричната крос-ентропия върху представителен корпус $D = \{x_i\}_{i=1}^{n}$:

$$H(\Pr_n[X] \| \hat{\Pr}_\theta[X]) = -\mathbb{E}_n[\log \hat{\Pr}_\theta[X]] = -\frac{1}{n}\sum_{i=1}^{n} \log \hat{\Pr}_\theta[X = x_i]$$

- Ако имаме корпус от двойки $D = \{(x_i, y_i)\}_{i=1}^{n}$ то бихме могли да разглеждаме емпиричната крос-ентропия на съвместното разпределения:

$$H(\Pr_n[X, Y] \| \hat{\Pr}_\theta[X, Y]) = -\mathbb{E}_n[\log \hat{\Pr}_\theta[X, Y]] = -\frac{1}{n}\sum_{i=1}^{n} \log \hat{\Pr}_\theta[x_i, y_i]$$

- Но ние не се интересуваме от съвместната вероятност $\hat{\Pr}_\theta[X, Y]$, а искаме да научим условната вероятност $\hat{\Pr}_\theta[Y \mid X]$.

Обучение на условен модел

- Нека ни е даден модел за $\hat{\Pr}_\theta[Y \mid X]$.
- Дефинираме съвместно разпределение: $\hat{\Pr}_\theta[X, Y] := \Pr_n[X]\ \hat{\Pr}_\theta[Y \mid X]$
- Замествайки във формулата за емпирична крос-ентропия за съвместното разпределение получаваме:
$$H(\Pr_n[X, Y] \| \hat{\Pr}_\theta[X, Y]) = -\mathbb{E}_n[\log \hat{\Pr}_\theta[X, Y]] =$$
$$= -\mathbb{E}_n[\log \Pr_n[X]] - \mathbb{E}_n[\log \hat{\Pr}_\theta[Y \mid X]] =$$
$$= H_X - \mathbb{E}_n[\log \hat{\Pr}_\theta[Y \mid X]]$$
- H_X не зависи от θ и следователно:
$$\arg\min_\theta H(\Pr_n[X, Y] \| \hat{\Pr}_\theta[X, Y]) = \arg\min_\theta -\mathbb{E}_n[\log \hat{\Pr}_\theta[Y \mid X]] =$$
$$= \arg\min_\theta -\frac{1}{n}\sum_{i=1}^{n} \log \hat{\Pr}_\theta[y_i \mid x_i] \quad .$$

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. **Условен езиков модел (10 мин)**
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Езиков модел

- За обучението на езиков модел представен с точна фамилия от локални разпределения $\{\Pr[w \mid w_1 w_2 \ldots w_k]\}_{w_1 w_2 \ldots w_k \in V^*}$ ни е необходим корпус от документи: $D = \{\mathbf{w}^{(i)}\}_{i=1}^{n}$.

- Обучението извършваме като минимизираме крос-ентропията:

$$H = -\frac{1}{\|D\|} \sum_{i=1}^{n} \sum_{j=1}^{|\mathbf{w}^{(i)}|+1} \log \Pr[\mathbf{w}_j^{(i)} \mid \mathbf{w}_1^{(i)}, \mathbf{w}_2^{(i)}, \ldots, \mathbf{w}_{j-1}^{(i)}]$$

- За целта използваме спускане по стохастичен градиент.

РНМ за представяне на езиков модел

- При входен текст $w_1 w_2 \ldots w_k$ с произволна дължина k, моделираме локалното вероятностно разпределение за следващата дума като:

- $\Pr[w \mid w_1 w_2 \ldots w_i] = \mathrm{softmax}(U\mathbf{h}_i)_w$, където $\mathbf{h}_j = g(\mathbf{h}_{j-1}, w_j)$, $\mathbf{h}_0$ фиксирано.

- Функцията g зависи от конкретното влагане и конкретната РНМ архитектура (LSTM, GRU, ...), брой слоеве и др.

Условен езиков модел

- Под условен езиков модел ще разбираме условно вероятностно разпределение $\mathbf{Pr}[\mathbf{w} \mid X]$, където $\mathbf{w} \in V^*$.

- Условен езиков модел може да бъде представен с точна фамилия от локални разпределения $\{\Pr[w \mid x; w_1 w_2 \ldots w_k]\}_{w_1 w_2 \ldots w_k \in V^*}$ за всяко $x \in X$.

- Ако моделираме експлицитно $\Pr[w \mid x; w_1 w_2 \ldots w_k]$, то ще можем да използваме модела за **условна генерация на текст.**

- Условното генериране на текст често се нарича **декодиране.**

Обучение на условен езиков модел

- За обучението ни е необходим корпус от двойки:
  $\mathbf{D} = \{(x^{(i)}, \mathbf{w}^{(i)}) \mid i = 1,2,\ldots,n\}$.

- Обучението ще извършим като минимизираме крос-ентропията на съвместното разпределение $\Pr[W, X]$ изразено чрез $\Pr[W \mid X]$:

$$H = -\frac{1}{\|\mathbf{D}\|}\sum_{i=1}^{n} \log \Pr[\mathbf{w}^{(i)} \mid x^{(i)}] =$$

$$= -\frac{1}{\|\mathbf{D}\|}\sum_{i=1}^{n}\sum_{j=1}^{|\mathbf{w}^{(i)}|+1} \log \Pr[\mathbf{w}_j^{(i)} \mid x^{(i)}; \mathbf{w}_1^{(i)}, \mathbf{w}_2^{(i)}, \ldots, \mathbf{w}_{j-1}^{(i)}]$$

- За целта отново ще използваме спускане по стохастичен градиент.

Реализиране на условен езиков модел с РНМ

- Прости подходи за реализиране на условен езиков модел с РНМ:

  1. $\Pr[w \mid x; w_1 w_2 \ldots w_i] = \mathrm{softmax}(U\mathbf{h}_i)_w$, където $\mathbf{h}_j = g(\mathbf{h}_{j-1}, w_j)$, и $\mathbf{h}_0 = f(x)$ (initial binding)

  2. $\Pr[w \mid x; w_1 w_2 \ldots w_i] = \mathrm{softmax}(U\mathbf{h}_i)_w$, където $\mathbf{h}_j = g(\mathbf{h}_{j-1}, f(w_j, x))$ (early binding)

  3. $\Pr[w \mid x; w_1 w_2 \ldots w_i] = \mathrm{softmax}(Uf(\mathbf{h}_i, x))_w$, където $\mathbf{h}_j = g(\mathbf{h}_{j-1}, w_j)$ (late binding)

- Съществуват значително по-софистицирани методи за реализиране на условен езиков модел с РНН. Ще разгледаме по-нататък архитектура с “Внимание”.

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. **Методи за генерация на текст / декодиране (20 мин)**
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Декодиране при условни езикови модели

- В приложенията, включващи условен езиков модел, обикновено целта е да намерим най-вероятната последователност $\mathbf{w}$ при дадено условие x.

- Т.е. задачата е да намерим последователността

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \sum_{j=1}^{|\mathbf{w}|+1} \log \Pr[\mathbf{w}_j \mid x; \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{j-1}]$$

- Горната задача в общия случай е неразрешима (съществуват безкраен брой последователности).

Метод на семплиране за генерация на текст с условен езиков модел

1. При дадено условие x започваме с празната последователност $\mathbf{w}^{(1)} = \varepsilon$.

2. Нека сме получили последователността $\mathbf{w}^{(i)} = \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}$. Намираме разпределението $\mathbf{Pr}[w \,|\, x; \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}]$.

3. Избираме $\hat{w}_i$ като семплираме с разпределението $\mathbf{Pr}[w \,|\, x; \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}]$. С така избрания елемент разширяваме последователността: $\mathbf{w}^{(i+1)} = \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}\hat{w}_i$.

4. Ако $\hat{w}_i$ е символът за край на последователност, то извеждаме така получената последователност и прекратяваме процедурата.

5. В противен случай отиваме в точка 2.

Особености на метода на семплиране за генерация на текст с условен езиков модел

- Този метод дава вариативност на резултата, която е желана при някои приложения, но е нежелана при други приложения.

- Няма изисквания за (лог-)вероятността $\sum_{j=1}^{|\mathbf{w}|+1} \log \Pr[\mathbf{w}_j \mid \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{j-1}]$ на получената последователност да е висока.

Метод на “Алчно декодиране” (Greedy decoding)

1. При дадено условие x започваме с празната последователност $\mathbf{w}^{(1)} = \varepsilon$.

2. Нека сме получили последователността $\mathbf{w}^{(i)} = \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}$. Намираме разпределението $\Pr[w \mid x; \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}]$.

3. Избираме елементът $\hat{w}_i$, който е най-вероятен при даденото разпределение. Т.е. $\hat{w}_i = \arg\max_w \Pr[w \mid x; \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}]$. С така избрания елемент разширяваме последователността: $\mathbf{w}^{(i+1)} = \hat{w}_1\hat{w}_2\ldots\hat{w}_{i-1}\hat{w}_i$.

4. Ако $\hat{w}_i$ е символът за край на последователност, то извеждаме така получената последователност и прекратяваме процедурата.

5. В противен случай отиваме в точка 2.

Особености на метода на алчно декодиране

- Целта ни е да получим последователност, която минимизира скоростта на крос-ентропия.
- Няма гаранция, че ще получим много вероятен резултат.
- Често се случва да изберем по средата на последователността даден елемент, поради което от там нататък генерираме по-малко вероятна последователност при даденото условие.
- Няма ли начин да се върнем назад — някакъв вид back-tracking?

Метод на търсене по лъча (Beam search)
Ширина на лъча β

---

1. При дадено услови x започваме с празната последователност $\mathbf{w}^{(1,1)} = \varepsilon$.

2. Нека на стъпка i сме получили не повече от β на брой последователности $\mathbf{w}^{(i,1)}, \mathbf{w}^{(i,2)}, \ldots, \mathbf{w}^{(i,\beta)}$, където $\mathbf{w}^{(i,j)} = \hat{w}_1^{(i,j)}\hat{w}_2^{(i,j)}\ldots\hat{w}_{i-1}^{(i,j)}$. За всяка от тях намираме разпределението $\Pr[w \mid x; \hat{\mathbf{w}}^{(i,j)}]$.

3. От разпределението $\Pr[w \mid x; \hat{\mathbf{w}}^{(i,j)}]$ намираме β на брой най-вероятни елемента $\hat{w}_1^{(i,j)}, \hat{w}_2^{(i,j)}, \ldots, \hat{w}_\beta^{(i,j)}$. С така избраните елементи разширяваме последователностите и получаваме β^2 кандидат последователности $\mathbf{w}^{(i,1)}\hat{w}_1^{(i,1)}, \ldots, \mathbf{w}^{(i,1)}\hat{w}_\beta^{(i,1)}, \ldots, \mathbf{w}^{(i,\beta)}\hat{w}_1^{(i,\beta)}, \ldots, \mathbf{w}^{(i,\beta)}\hat{w}_\beta^{(i,\beta)}$.

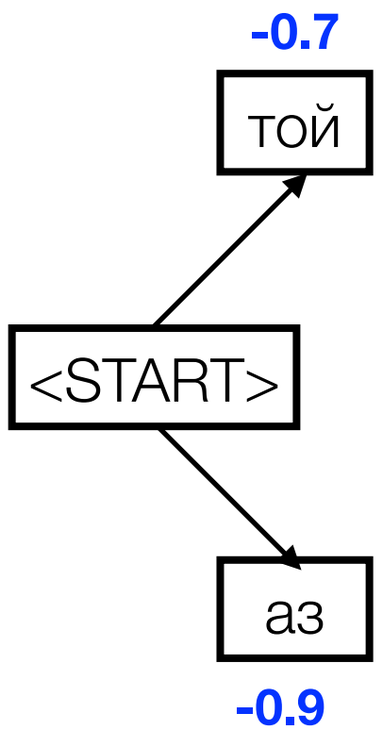
4. От получените β^2 кандидат последователности намираме β с най-висока вероятност $\sum_{k=1}^{i} \log \Pr[\hat{w}^{(i,j)} \mid x; \hat{w}_1^{(i,j)}\hat{w}_2^{(i,j)}\ldots\hat{w}_{k-1}^{(i,j)}]$. Получените β на брой последователности означаваме като $\mathbf{w}^{(i+1,1)}, \mathbf{w}^{(i+1,2)}, \ldots, \mathbf{w}^{(i+1,\beta)}$.

5. От списъка с последователностите премахваме и извеждаме, завършващите със символът за край на последователност. Ако сме получили достатъчно изведени последователности, избираме най-добрия и прекратяваме процедурата.

6. В противен случай отиваме в точка 2.

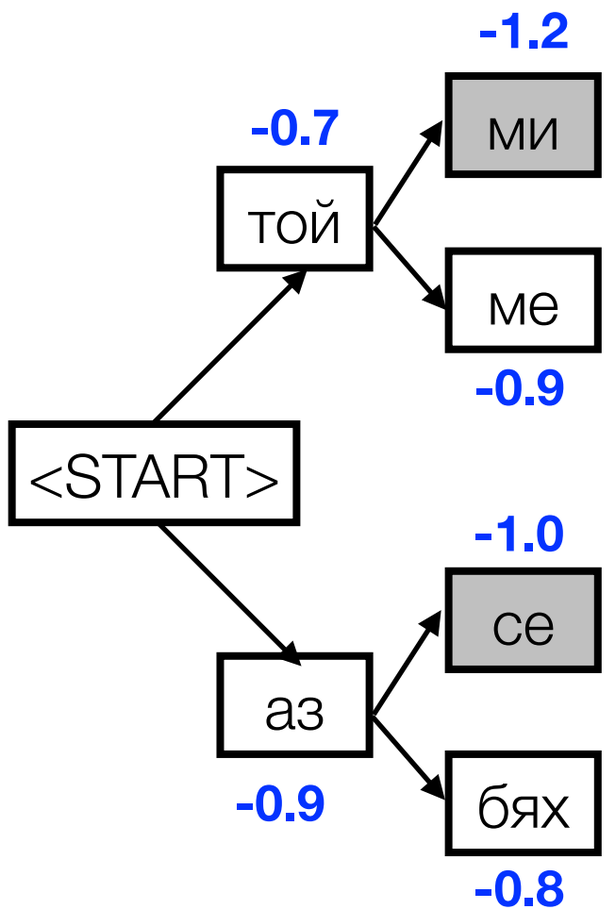
Пример за търсене по лъча ($\beta = 2$)

Пример за търсене по лъча ($\beta = 2$)



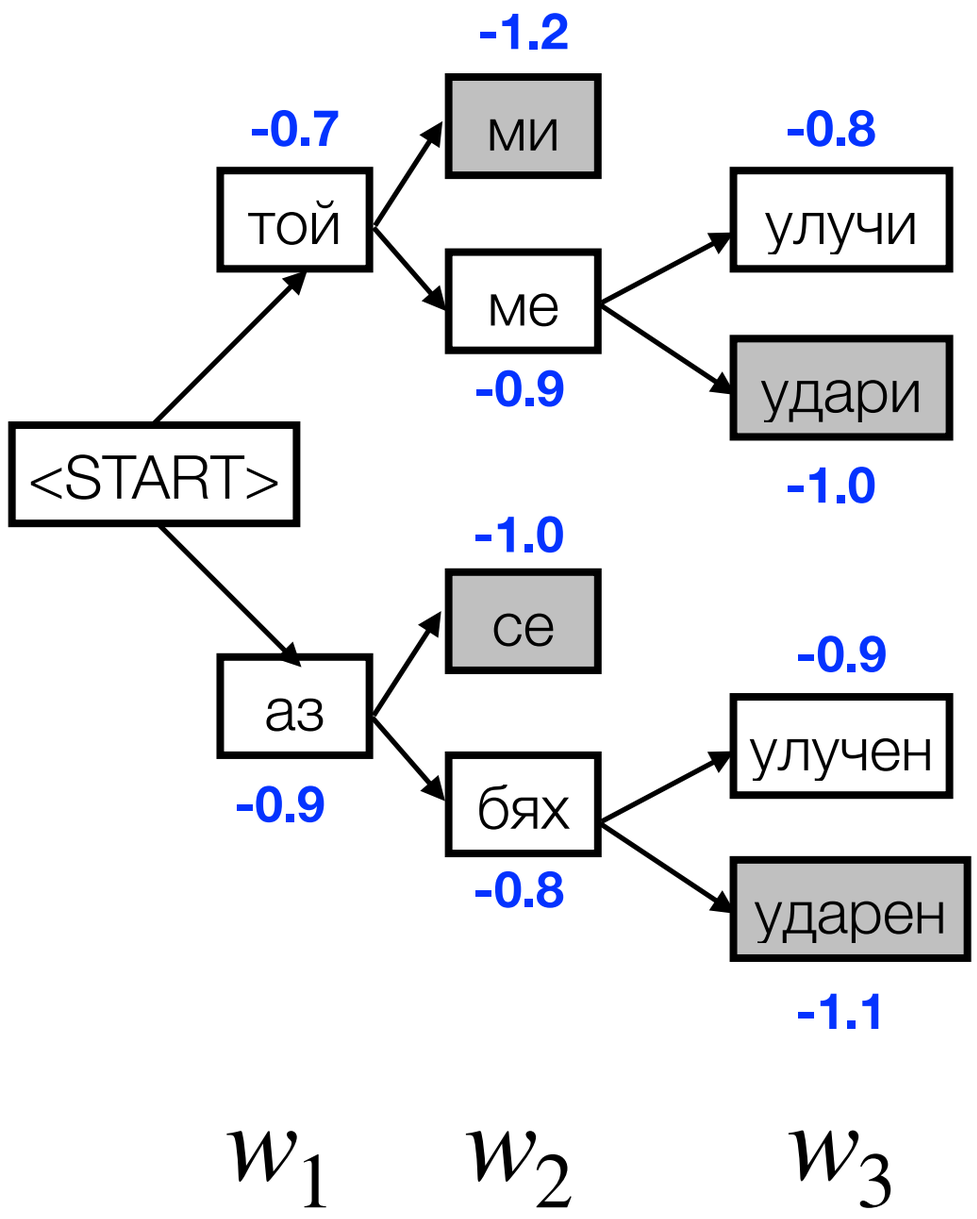
w_1

Пример за търсене по лъча ($\beta = 2$)

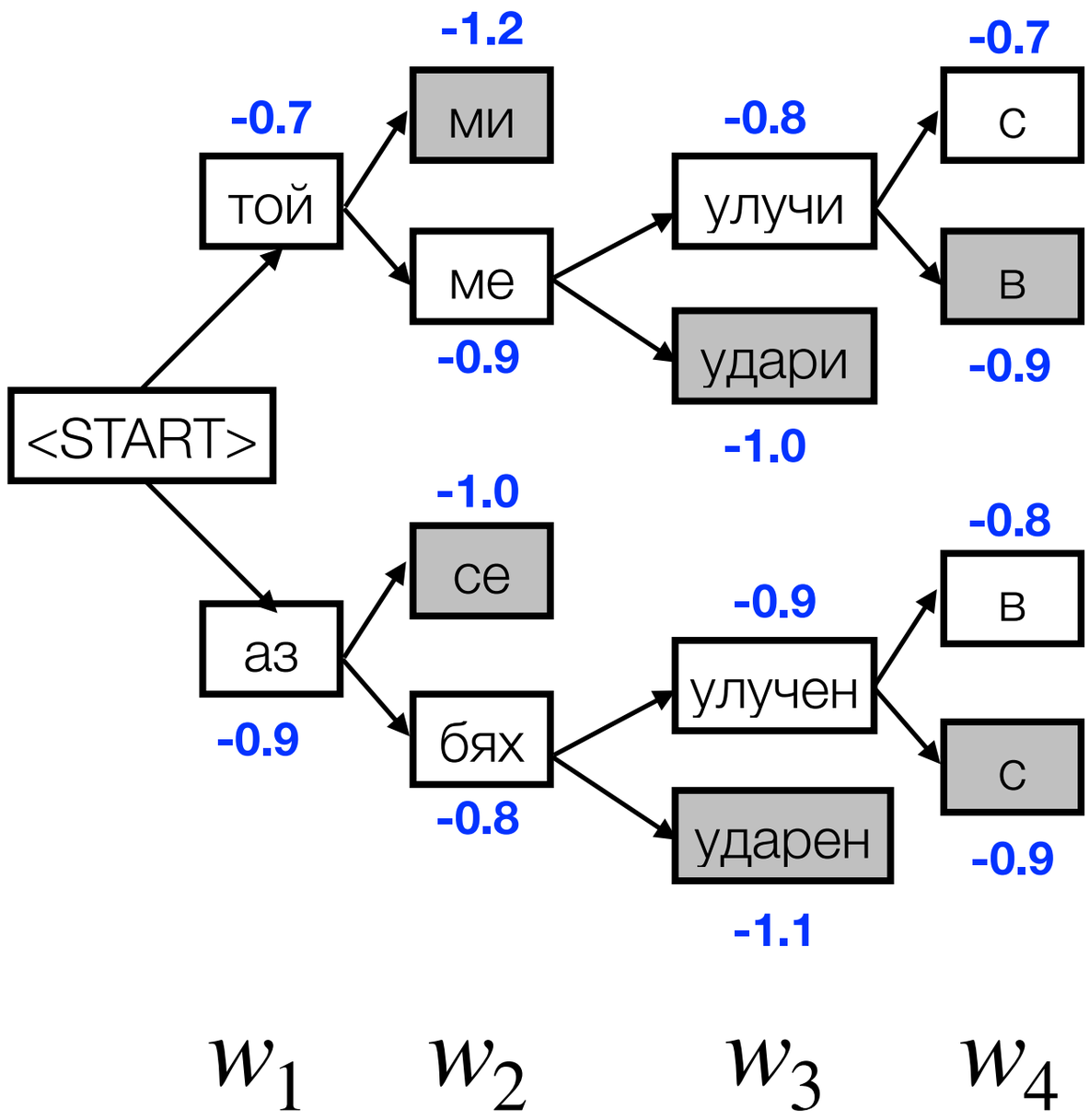


w_1 w_2

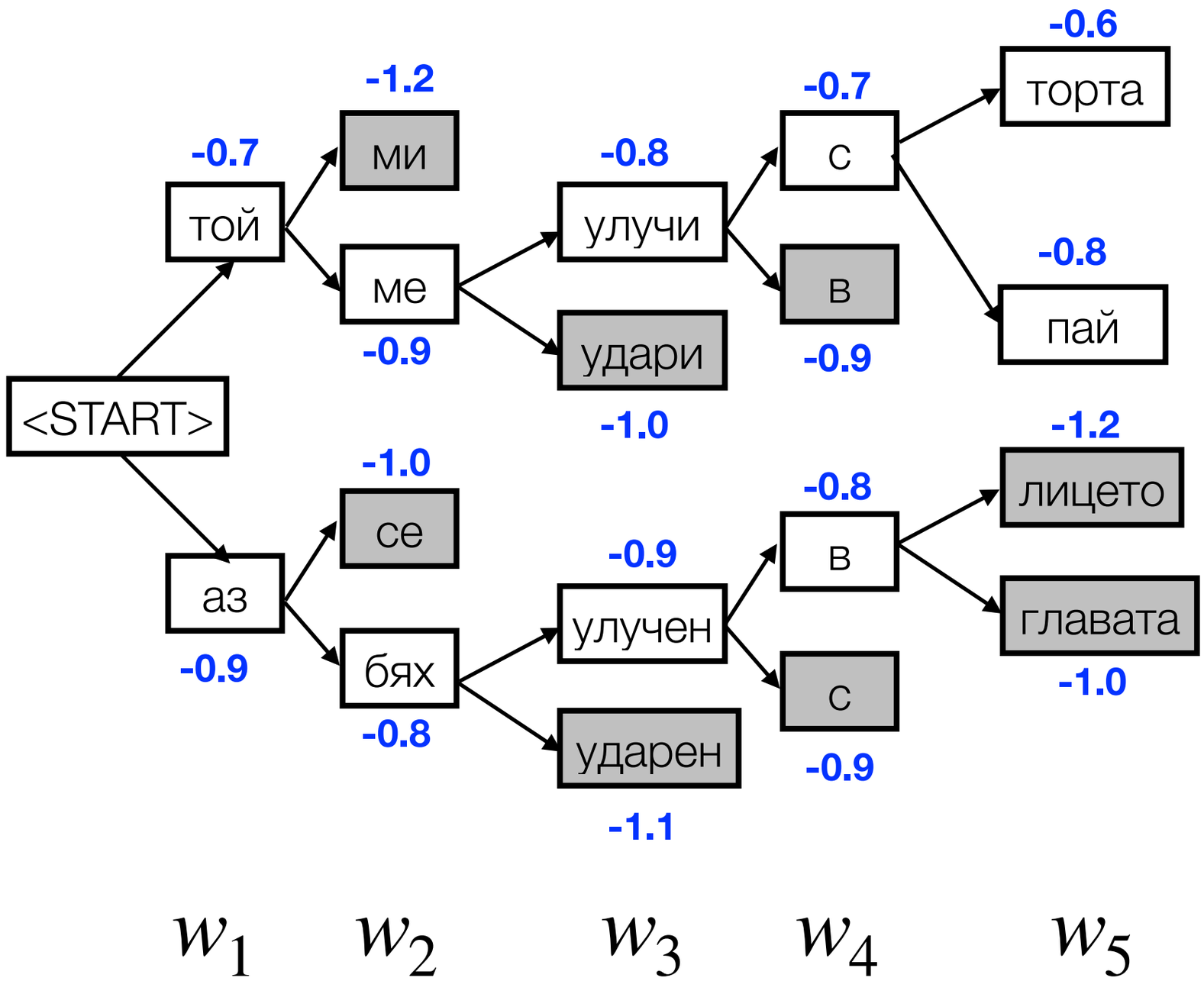
Пример за търсене по лъча ($\beta = 2$)



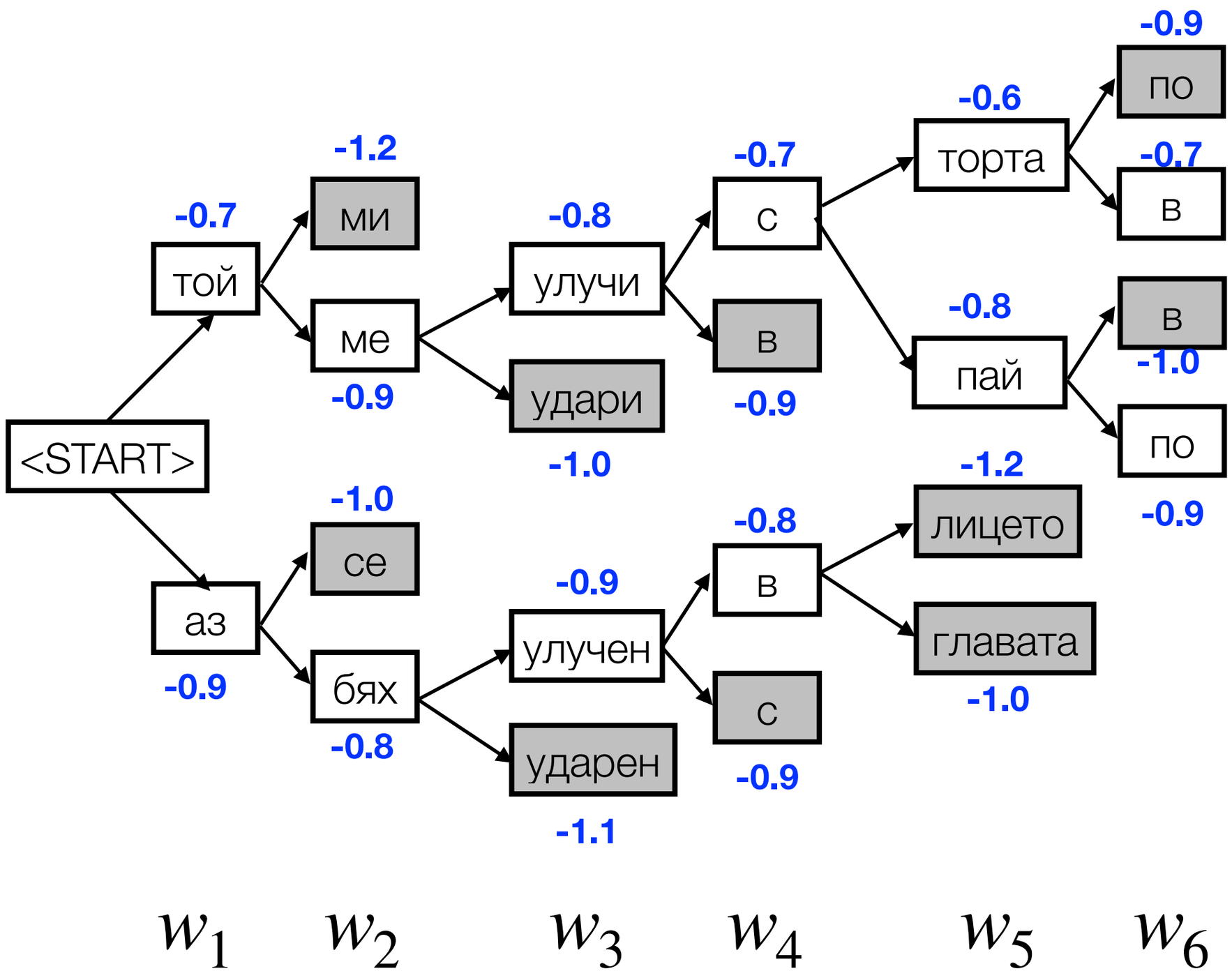
Пример за търсене по лъча ($\beta = 2$)



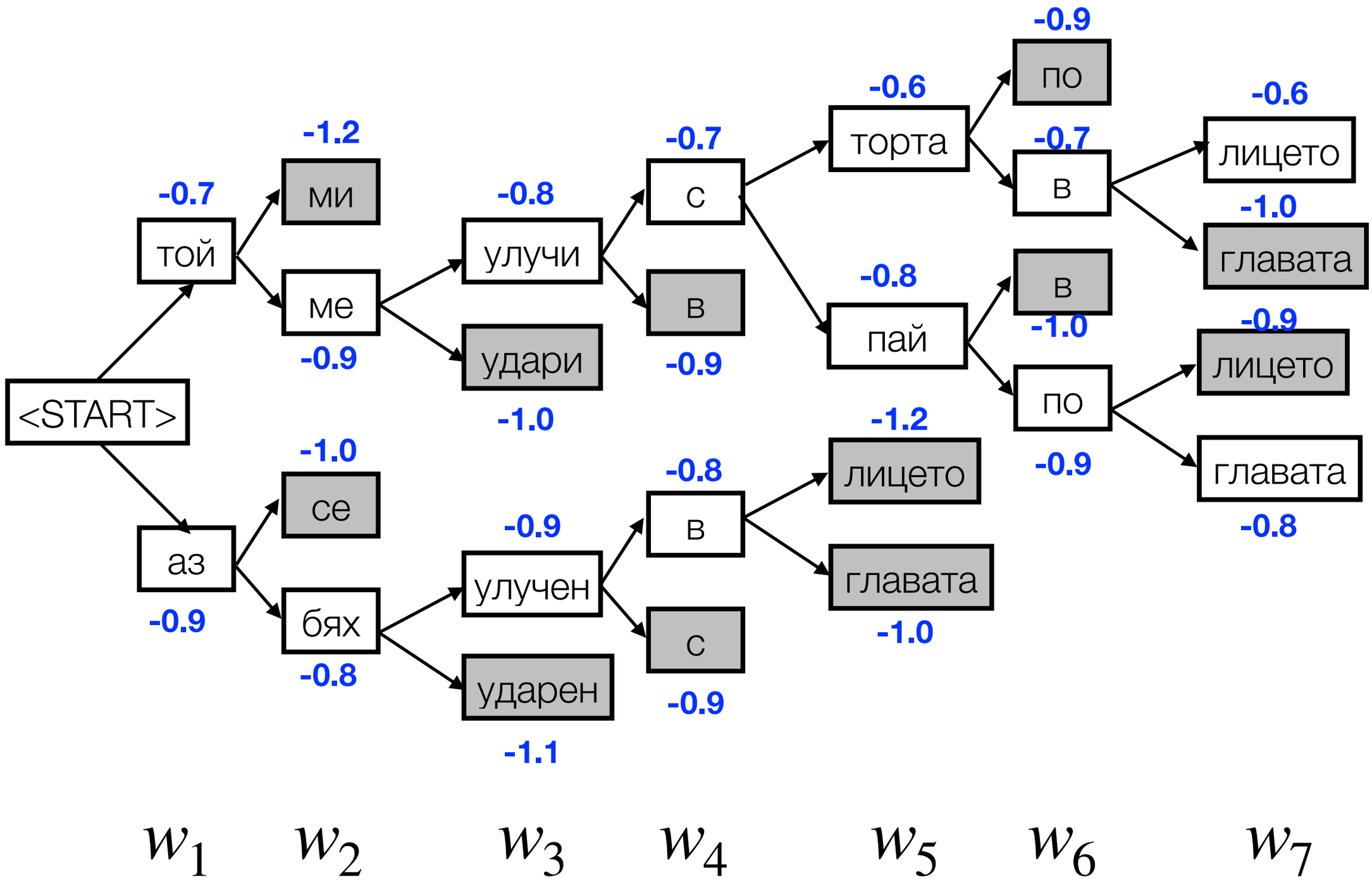
Пример за търсене по лъча ($\beta = 2$)



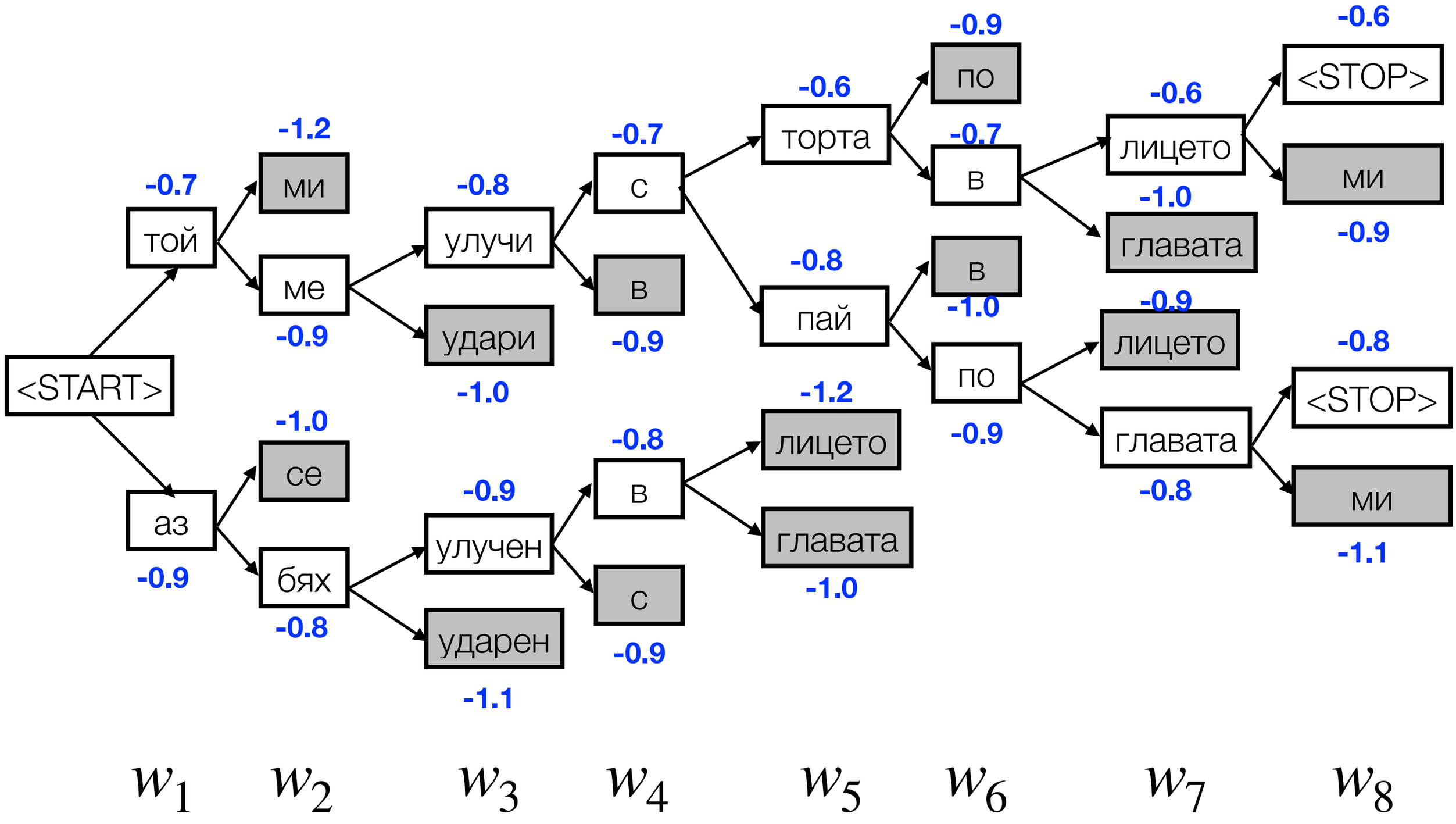
Пример за търсене по лъча ($\beta = 2$)



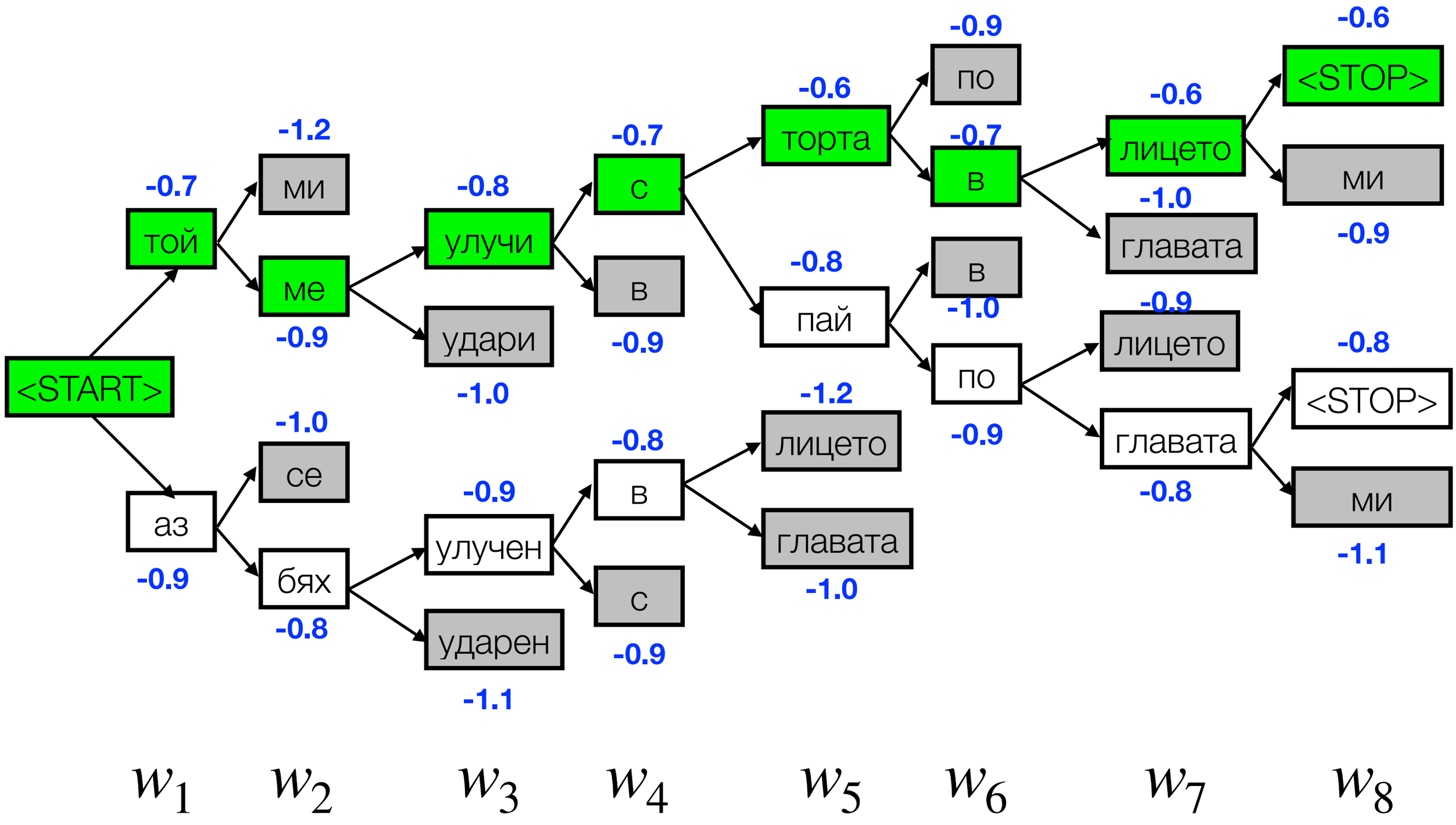
Пример за търсене по лъча ($\beta = 2$)



Пример за търсене по лъча ($\beta = 2$)



Пример за търсене по лъча ($\beta = 2$)



Особености при търсенето по лъча

- Не гарантира намирането на най-вероятната последователност.
- При добър избор на β резултатът е по-добър от алчния избор.
- Представя евристичен компромис между изчислителна ефективност и пълнота и коректност.
- На практика параметърът β се избира в порядък 3-500 с напасване.

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. **Приложения на генерация на текст с езиков модел (5 мин)**
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Приложения на условната генерация на текст

Условие x	**Изходен текст w**
Описание на задача	Изходен код на програма
Автор	Текст със стила на автора
Тема	Статия по темата
Изречение на английски	Превод на български
Снимка	Описание на снимката
Статия	Резюме
Запис на реч	Транскрипция
Въпрос + статия	Отговор

Модел “Последователност към последователност”

- При тези модели входна последователност от елементи се проебразува в изходна последователност.

- Примери:
  - Машинен превод от един език на друг
  - Автоматично разпознаване на реч
  - Генериране на резюме на статия

Модел “Последователност към последователност”

- Условието е последователност: $\mathbf{x} = x_1 x_2 \ldots x_l$.

- Търсим последователност $\mathbf{w} = w_1 w_2 \ldots w_k$, така че

$$\mathbf{w} = \arg\max_{\mathbf{w}} \Pr[\mathbf{w} \mid \mathbf{x}] = \arg\max_{\mathbf{w}} \prod_{i=1}^{k} \Pr[w_i \mid \mathbf{x}; w_1 w_2 \ldots w_{i-1}] =$$

$$= \arg\max_{\mathbf{w}} \sum_{i=1}^{k} \log \Pr[w_i \mid \mathbf{x}; w_1 w_2 \ldots w_{i-1}]$$

- Wu et al. (2016): Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. https://arxiv.org/abs/1609.08144)

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. **Архитектура енкодер-декодер за невронен машинен превод (15 мин)**
7. Архитектура за “внимание” (Attention) (20 мин)
8. Оценяване на резултат от машинен превод (5 мин)

Статистически (преди невронен) машинен превод

- $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \Pr[\mathbf{y} \mid \mathbf{x}] = \arg\max_{\mathbf{y}} \frac{\Pr[\mathbf{x} \mid \mathbf{y}] \Pr[\mathbf{y}]}{\Pr[\mathbf{x}]} = \arg\max_{\mathbf{y}} \Pr[\mathbf{x} \mid \mathbf{y}] \Pr[\mathbf{y}]$
- $\Pr[\mathbf{y}]$ е езиков модел на целевия езикът
- $\Pr[\mathbf{x} \mid \mathbf{y}]$ е преводен модел — ще се стремим да разбием поелементно. За целта ни е необходимо подравняване (alignment) a.
- Подравняването може да разгледаме като функция, която на дадена позиция в изходната последователност съпоставя позиция в целевата последователност. (Има по-добри модели за подравняване.)
- $\Pr[\mathbf{x} \mid \mathbf{y}] = \sum_{a} \Pr[\mathbf{x}, a \mid \mathbf{y}] = \sum_{a} \prod_{i} \Pr[a(i) \mid \mathbf{x}, \mathbf{y}] \Pr[\mathbf{x}_i \mid \mathbf{y}_{a(i)}]$
- Търсим $\hat{\mathbf{y}} = \arg\max_{\mathbf{y},a} \Pr[\mathbf{y}] \prod_{i} \Pr[a(i) \mid \mathbf{x}, \mathbf{y}] \Pr[\mathbf{x}_i \mid \mathbf{y}_{a(i)}]$

Пример за подравняване

	Аз	бях	на	кино
I				
have				
been				
to				
the				
cinema				

Невронен машинен превод — пробив 2014 г.

Sutskever, Vinyals and Le (2014): Sequence to Sequence Learning with Neural Networks
https://arxiv.org/abs/1409.3215v3

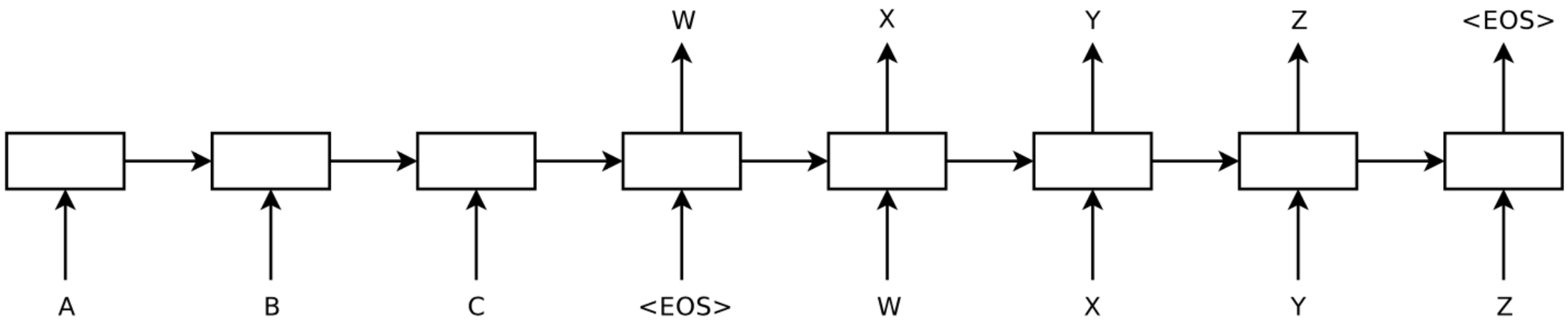
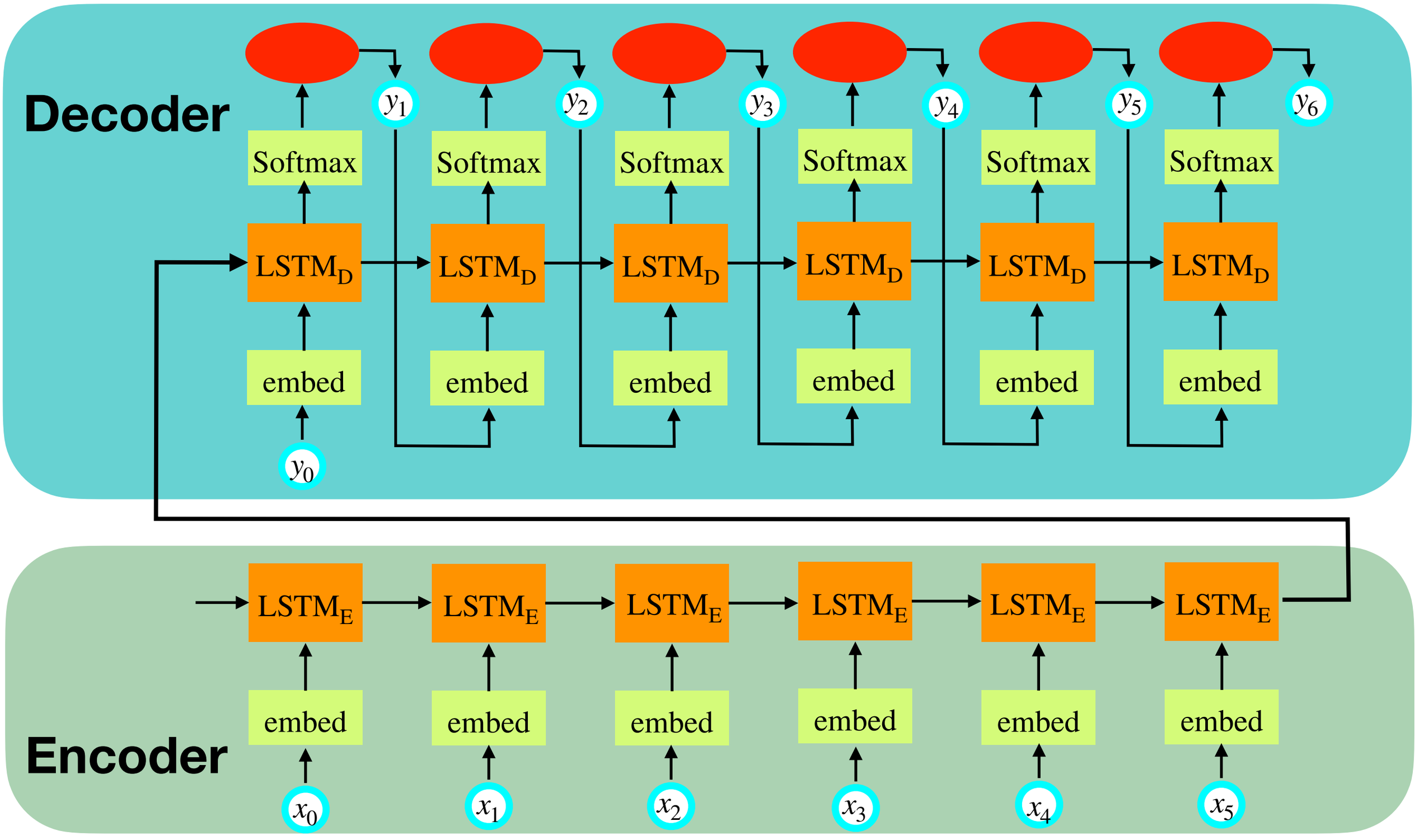
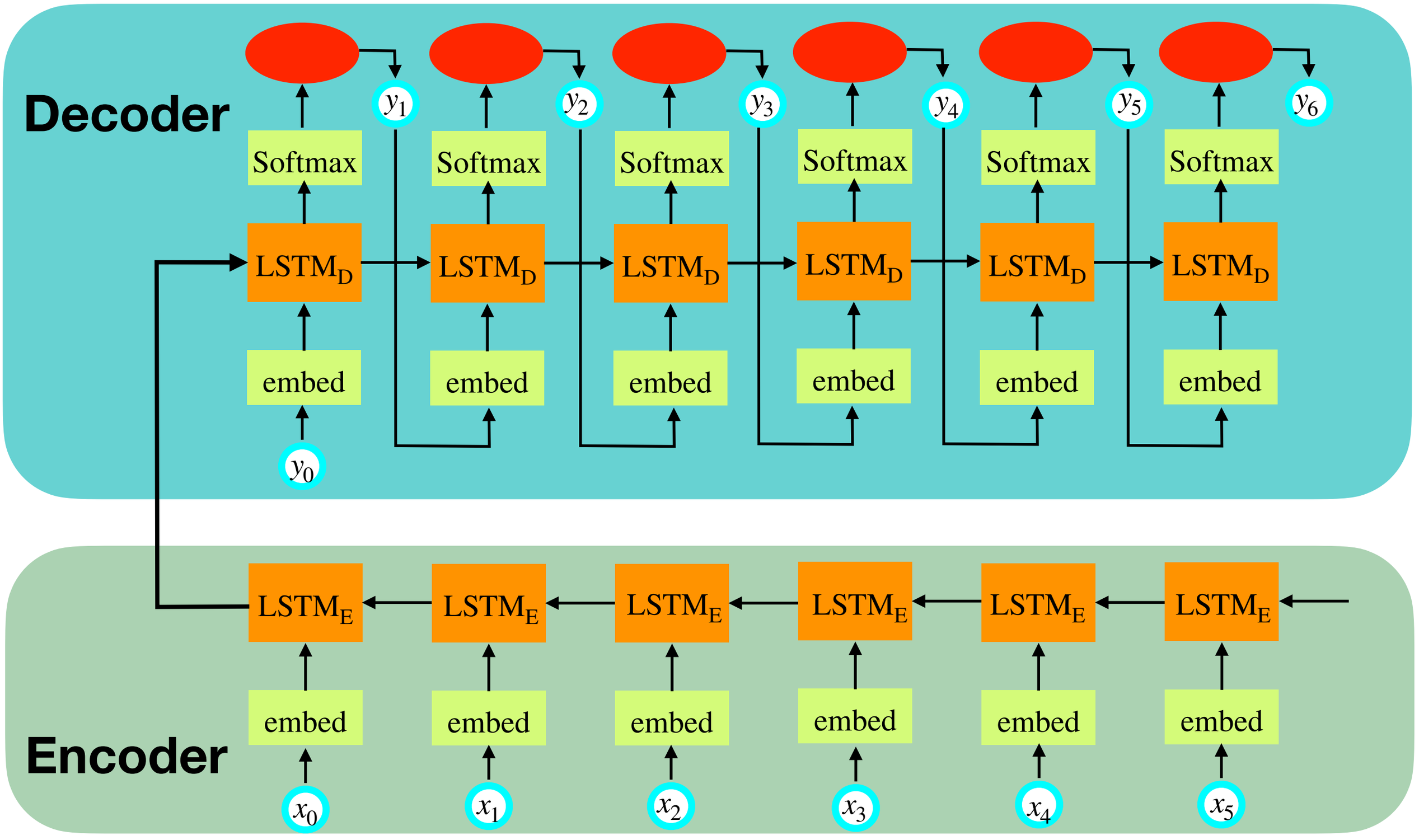


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

Невронен машинен превод — пробив 2014 г.



Невронен машинен превод — пробив 2014 г.



Невронен машинен превод — пробив 2014 г.

- В модела на Sutskever входният текст се кодира от LSTM рекурентна невронна мрежа.
- Последният скрит вектор и състояние от LSTM мрежата на енкодера се подава като начален скрит вектор и състояние на LSTM мрежата на декодера.
- Това съответства на първия подход за реализиране на условен езиков модел с PHM (initial binding).
- Експериментите описани в статията показват малко по-добри резултати при обръщане на входната последователност.
- **Проблем**: Размерността на скрития вектора, кодиращ входната последователност не зависи от дължината му.

Prof. Ray Mooney: “You can't cram the meaning of a whole %&!$# sentence into a single $&!# vector!”*

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. **Архитектура за “внимание” (Attention) (20 мин)**
8. Оценяване на резултат от машинен превод (5 мин)

Реализиране на архитектура за “внимание” — Attention

- Резултатът от кодирането на входната последователност не е само последния скрит вектор, а последователността от всички скрити вектори по пътя — $\mathbf{h}_1^E, \mathbf{h}_2^E, \ldots, \mathbf{h}_l^E$. Размерът зависи от дължината на последователността.

- Архитектурата на декодера очаква вход с фиксиран размер.

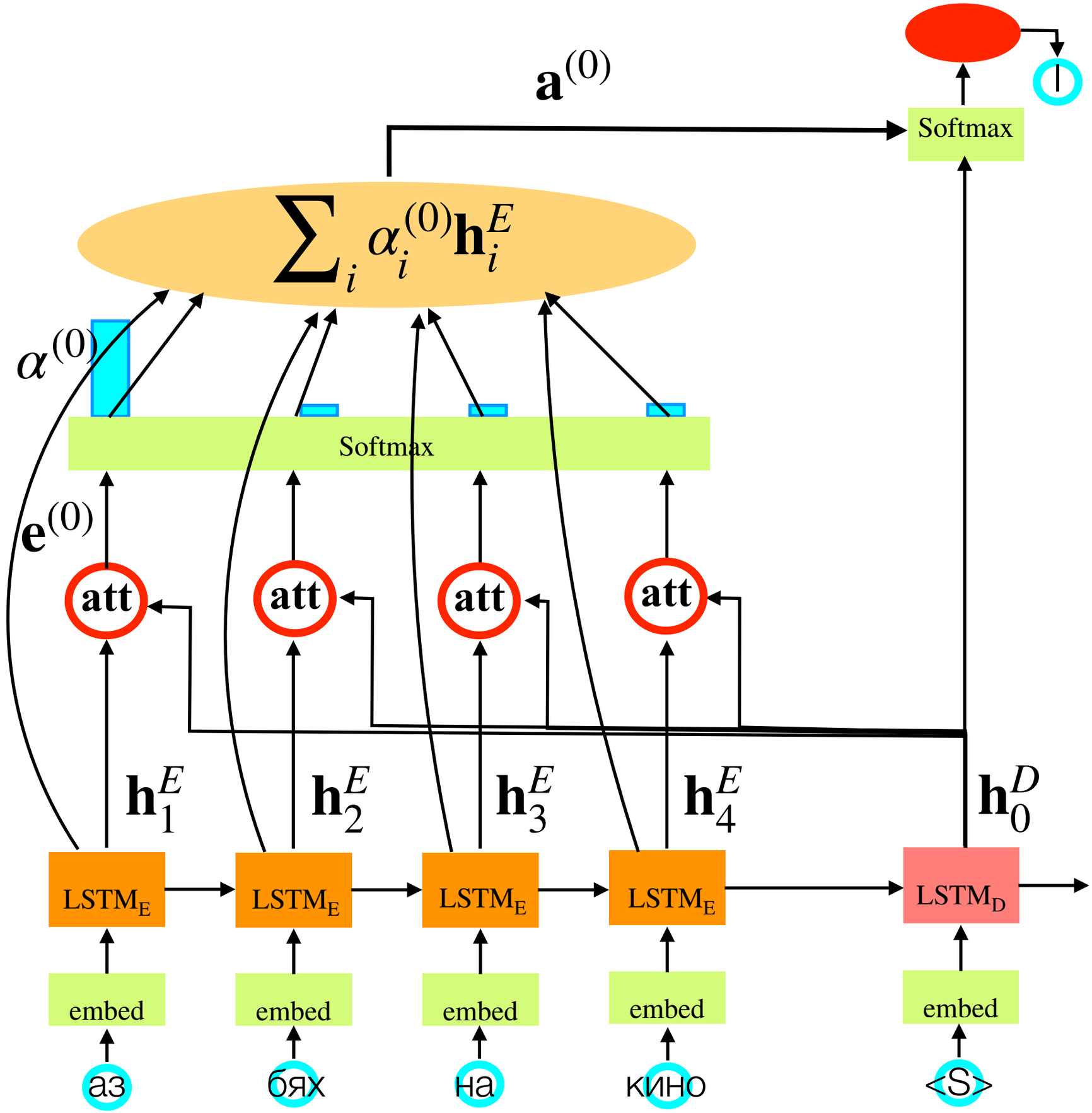
- В дадена позиция резултатът от декодирането следва да зависи повече от скритията вектор, който е около съответната подравнена позиция при кодирането на входната последователност.

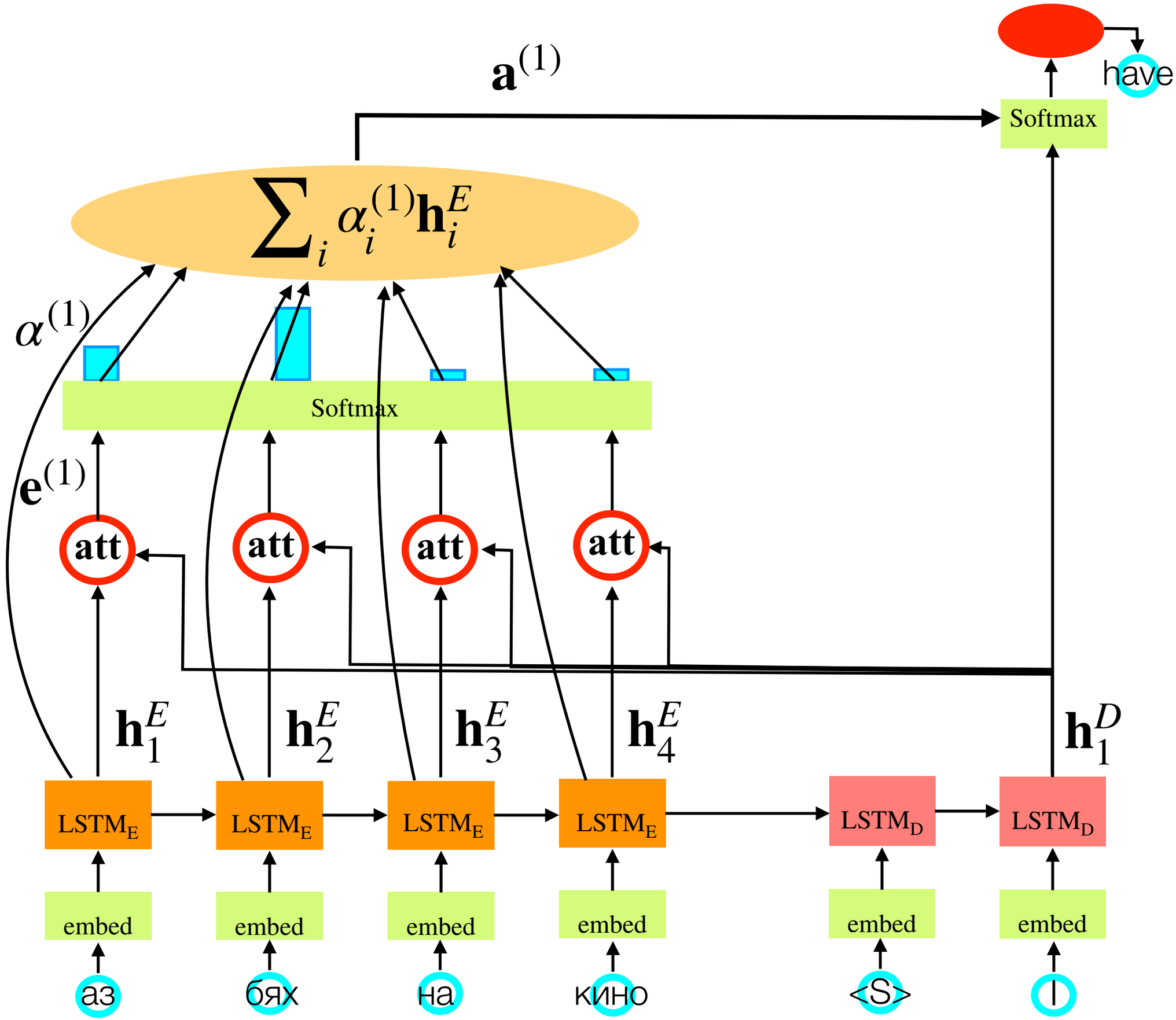
- Ще реализираме архитектура на внимание за моделирането на подравняване.

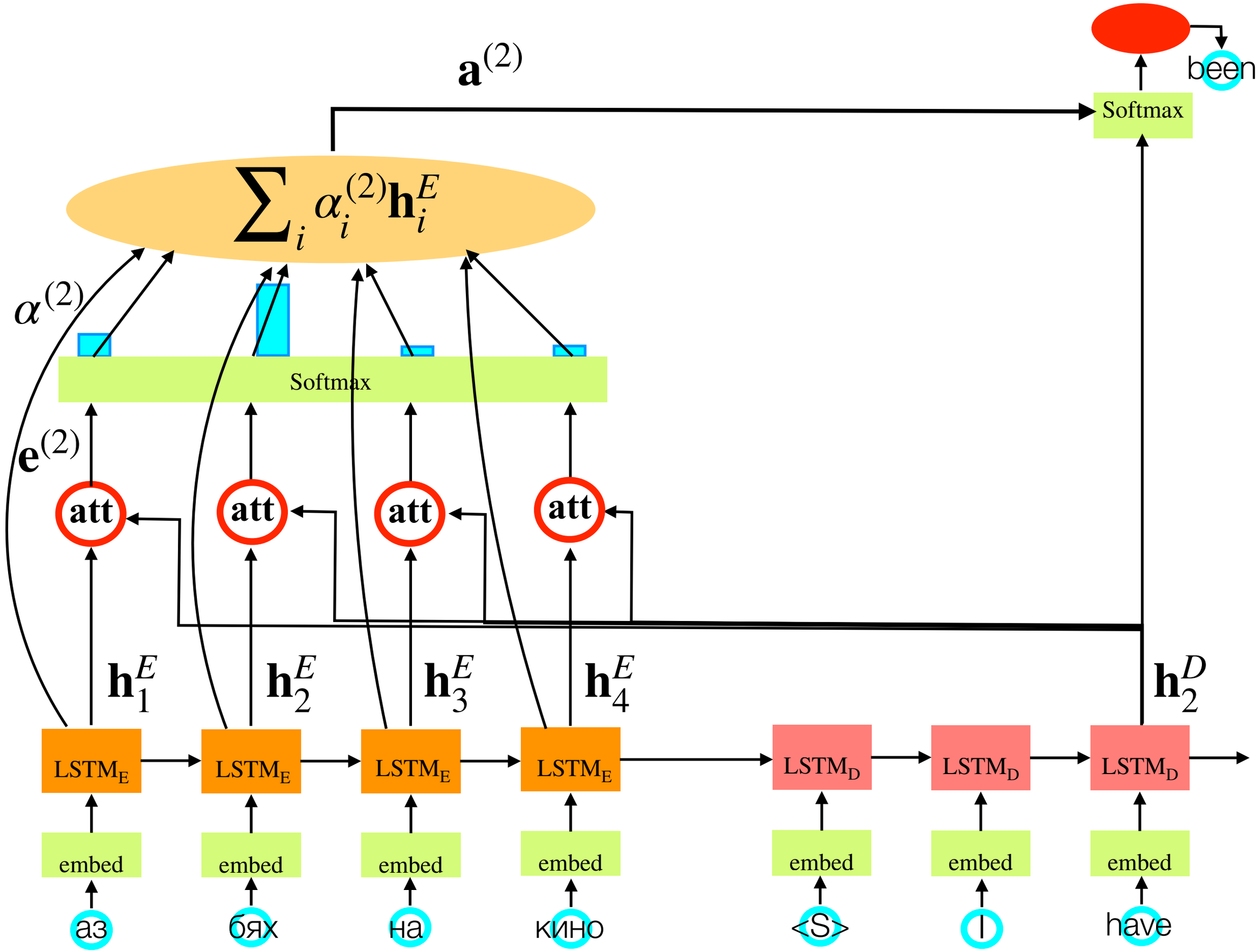
Bahdanau, Cho and Bengio (2014): Neural Machine Translation by Jointly Learning to Align and Translate,
https://arxiv.org/abs/1409.0473

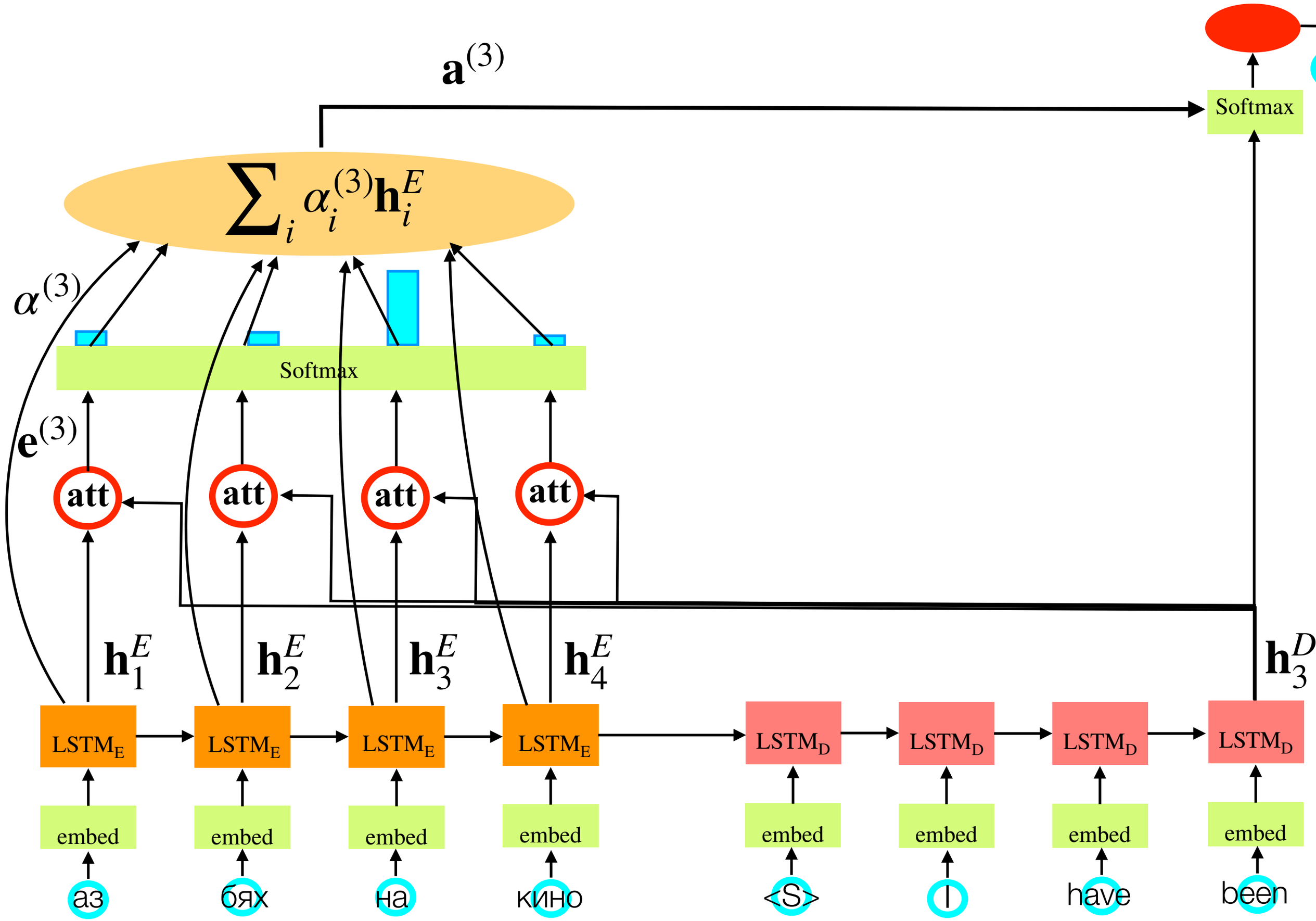
Реализиране на внимание

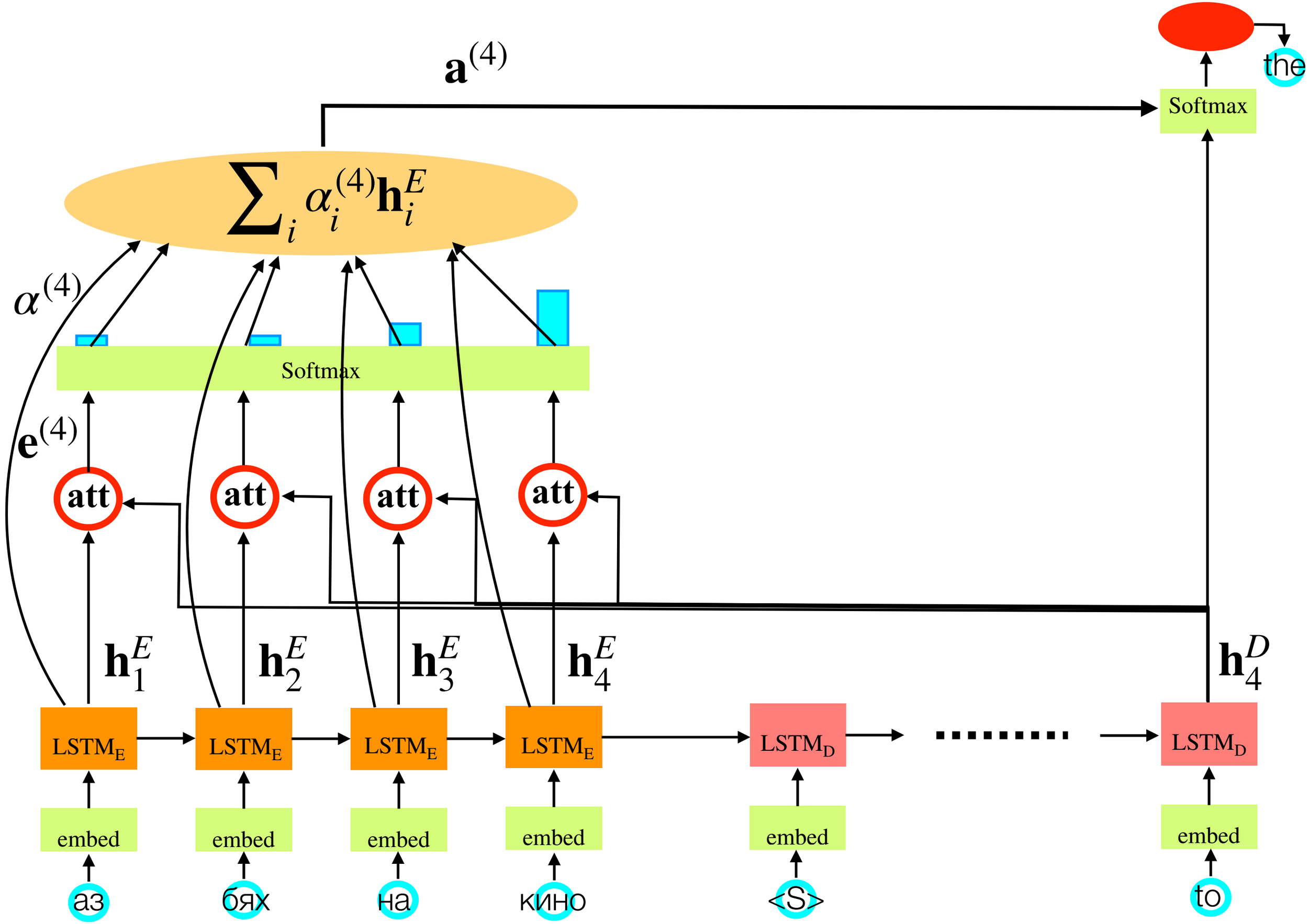
- Нека входната последователност $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l$ е кодирана в последователността от скрити вектори $\mathbf{h}_1^E, \mathbf{h}_2^E, \ldots, \mathbf{h}_l^E$.
- Нека сме стигнали при декодиране до позиция j, на която сме получили скрит вектор $\mathbf{h}_j^D$.
- Създаваме вектора “размер на внимание” (attention score) в позиция j:
  $\mathbf{e}^{(j)} = [\text{att}(\mathbf{h}_j^D, \mathbf{h}_1^E), \text{att}(\mathbf{h}_j^D, \mathbf{h}_2^E), \ldots, \text{att}(\mathbf{h}_j^D, \mathbf{h}_l^E)] \in \mathbb{R}^l$
- Намираме теглата в позиция j: $\alpha^{(j)} = \text{softmax}(\mathbf{e}^{(j)}) \in \mathbb{R}^l$
- Претегляйки скритите вектори при декодиране с теглата получаваме вектора на внимание в позиция j: $\mathbf{a}^{(j)} = \sum_{i=1}^{l} \alpha_i^{(j)} \mathbf{h}_i^E \in \mathbb{R}^h$
- Накрая конкатенираме вектора на внимание с скрития вектор при декодиране за намирането на вероятността за следващата дума: $[\mathbf{a}^{(j)}, \mathbf{h}_j^D] \in \mathbb{R}^{2h}$

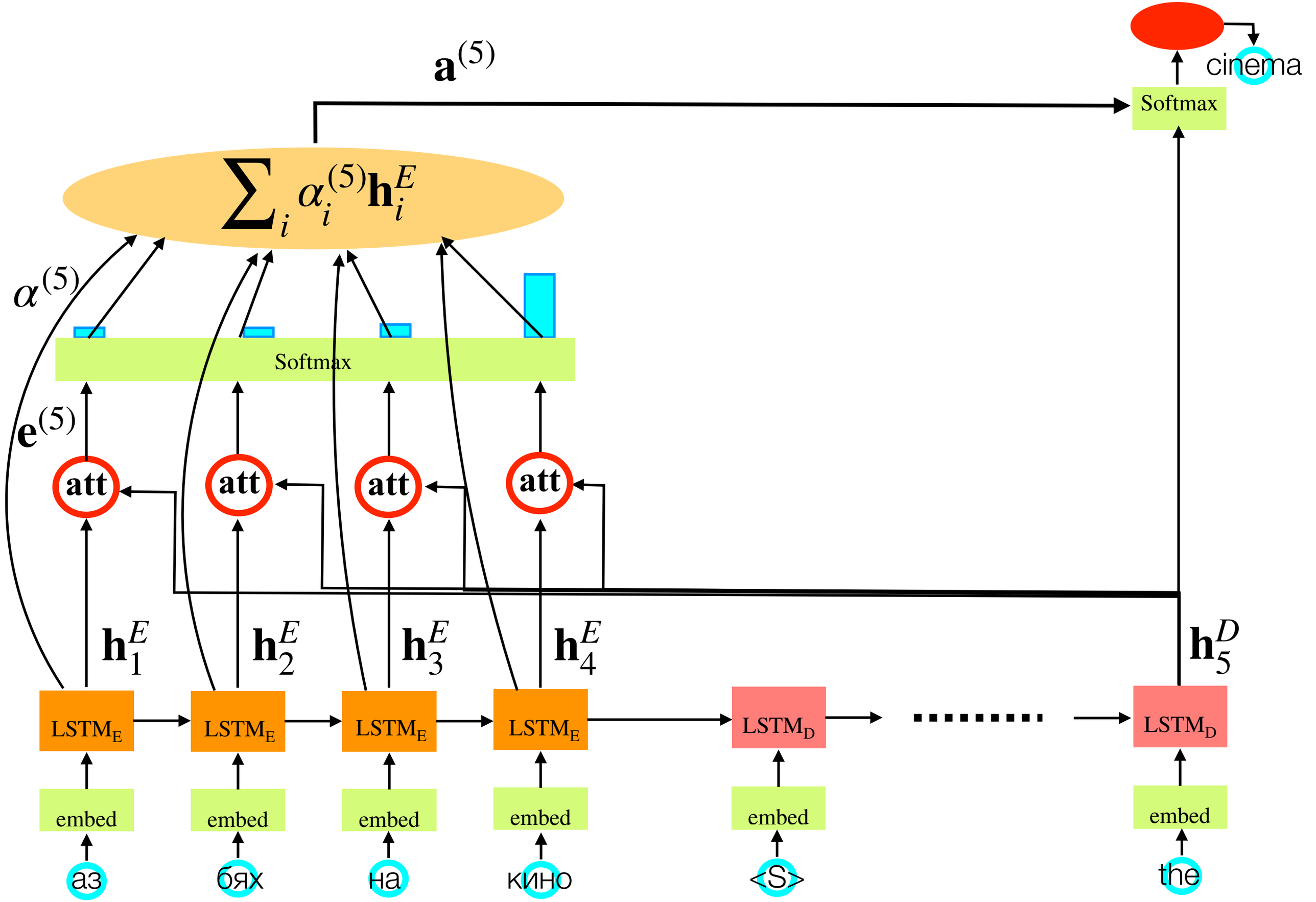

$\mathbf{a}^{(0)}$
Softmax
$\sum_i \alpha_i^{(0)} \mathbf{h}_i^E$
$\alpha^{(0)}$
Softmax
$\mathbf{e}^{(0)}$
att
att
att
att
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_0^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>

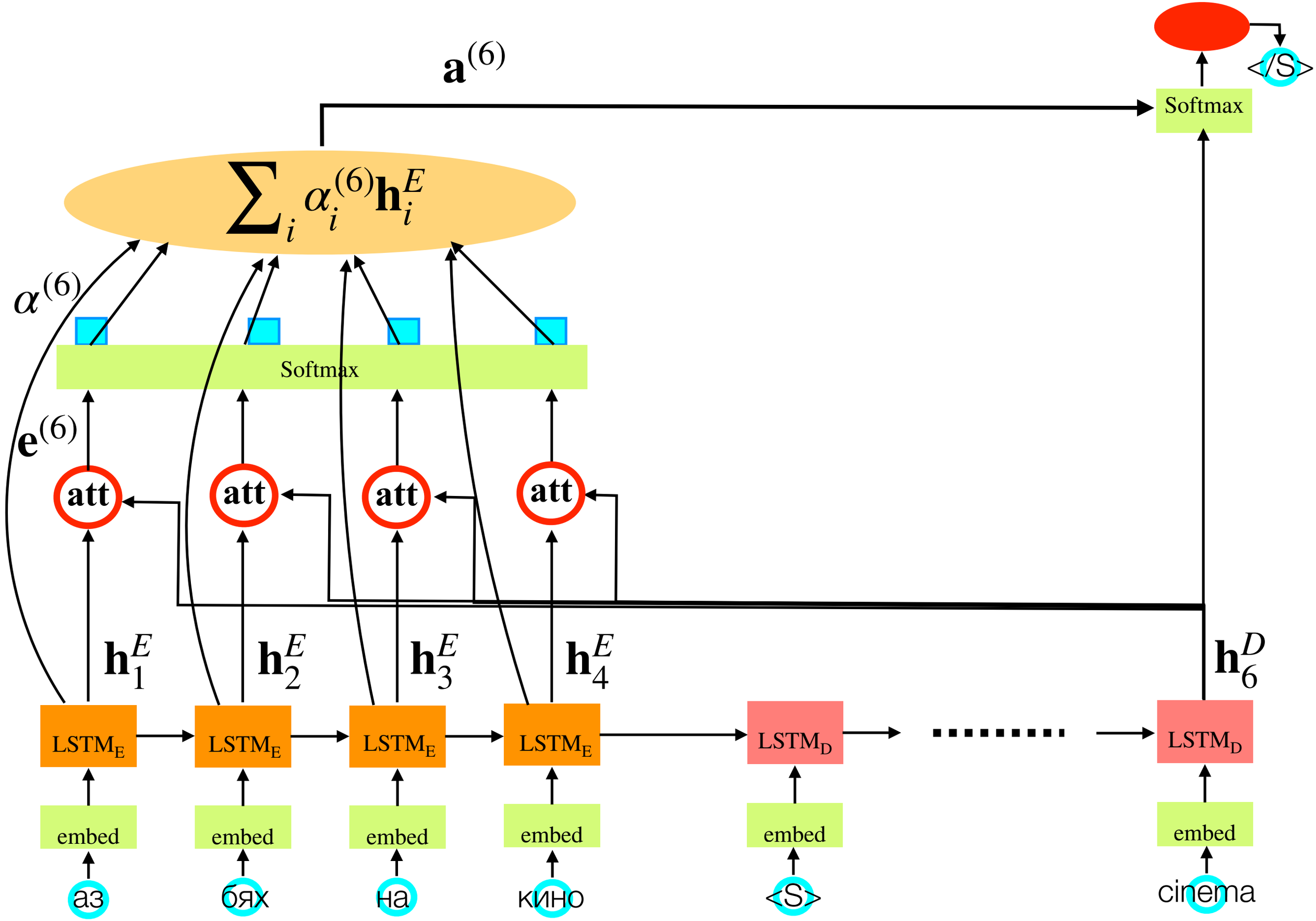

$\mathbf{a}^{(1)}$
$\sum_i \alpha_i^{(1)} \mathbf{h}_i^E$
$\alpha^{(1)}$
Softmax
$\mathbf{e}^{(1)}$
att
att
att
att
Softmax
have
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_1^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
I


$\mathbf{a}^{(2)}$
$\sum_i \alpha_i^{(2)} \mathbf{h}_i^E$
$\alpha^{(2)}$
Softmax
$\mathbf{e}^{(2)}$
att
att
att
att
Softmax
been
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_2^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
I
have


$\mathbf{a}^{(3)}$
$\sum_i \alpha_i^{(3)} \mathbf{h}_i^E$
$\alpha^{(3)}$
Softmax
Softmax
$\mathbf{e}^{(3)}$
att
att
att
att
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_3^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
I
have
been


$\mathbf{a}^{(4)}$
$\sum_i \alpha_i^{(4)} \mathbf{h}_i^E$
$\alpha^{(4)}$
Softmax
$\mathbf{e}^{(4)}$
att
att
att
att
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_4^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
to
Softmax
the


$\mathbf{a}^{(5)}$
$\sum_i \alpha_i^{(5)} \mathbf{h}_i^E$
$\alpha^{(5)}$
Softmax
$\mathbf{e}^{(5)}$
att
att
att
att
Softmax
cinema
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_5^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
the


$\mathbf{a}^{(6)}$
$\sum_i \alpha_i^{(6)} \mathbf{h}_i^E$
$\alpha^{(6)}$
Softmax
$\mathbf{e}^{(6)}$
att
att
att
att
Softmax
</S>
$\mathbf{h}_1^E$
$\mathbf{h}_2^E$
$\mathbf{h}_3^E$
$\mathbf{h}_4^E$
$\mathbf{h}_6^D$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_E$
$LSTM_D$
$LSTM_D$
embed
embed
embed
embed
embed
embed
аз
бях
на
кино
<S>
cinema

Варианти за функцията на внимание

$$\mathbf{e}^{(j)} = [\text{att}(\mathbf{h}_j^D, \mathbf{h}_1^E), \text{att}(\mathbf{h}_j^D, \mathbf{h}_2^E), \ldots, \text{att}(\mathbf{h}_j^D, \mathbf{h}_l^E)] \in \mathbb{R}^l$$

1. Най-прост вариант — скаларно произведение: $\text{att}(\mathbf{h}_j^D, \mathbf{h}_i^E) = (\mathbf{h}_j^D)^\top \mathbf{h}_i^E$. В този случай е необходимо размерностите на скритите вектори на енкодера и декодера да съвпадат.

2. Мултипликативно внимание:
   $\text{att}(\mathbf{h}_j^D, \mathbf{h}_i^E) = (\mathbf{h}_j^D)^\top W \mathbf{h}_i^E$, където $W \in \mathbb{R}^{h_D \times h_E}$ е матрица с тегла.

3. Адитивно внимание:
   $\text{att}(\mathbf{h}_j^D, \mathbf{h}_i^E) = \mathbf{v}^\top \tanh(W_1 \mathbf{h}_j^D + W_2 \mathbf{h}_i^E)$, където $W_1 \in \mathbb{R}^{d \times h_D}$, $W_2 \in \mathbb{R}^{d \times h_E}$ са матрици с тегла, $\mathbf{v} \in \mathbb{R}^d$ е вектор с тегла и d е метапараметър.

Експериментите показват, че адитивният вариант 3 дава малко по-добри резултати, но е изчислително по-тежък. Виж:
Britz et al, 2017: Massive Exploration of Neural Machine Translation Architectures,
https://arxiv.org/abs/1703.03906

План на лекцията

1. Формалности за курса (5 мин)
2. Обучение на условен модел (10 мин)
3. Условен езиков модел (10 мин)
4. Методи за генерация на текст / декодиране (20 мин)
5. Приложения на генерация на текст с езиков модел (5 мин)
6. Архитектура енкодер-декодер за невронен машинен превод (15 мин)
7. Архитектура за “внимание” (Attention) (20 мин)
8. **Оценяване на резултат от машинен превод (5 мин)**

Методи за оценяване на качеството на превода

1. Сравняване на крос-ентропията — лесно за изчисляване но не е пряко свързано с качеството на превода — трудно може да се интерпретира.

2. Оценяване от експерт — прецизно но много скъпо за реализиране.

3. Сравнение на общи k-грами с реферативен професионален превод — сравнително лесно за реализиране и дава до известна степен интерпретируема оценка за превода.

Papineni et al, ACL-2002: BLEU: a Method for Automatic Evaluation of Machine Translation,
https://www.aclweb.org/anthology/P02-1040.pdf

BLEU (Bilingual Evaluation Understudy)

BLEU е претеглено геометрично средно на прецизността на k-грамите с фактор за наказване на твърде къси преводи:

$$\mathrm{Bleu}_4 = \exp(0.5 \log P_1 + 0.25 \log P_2 + 0.125 \log P_3 + 0.125 \log P_4 - \max(\frac{|\bar{\mathbf{y}}|}{|\mathbf{y}|} - 1{,}0)),$$

- P_1 е прецизността на 1-грамите — процентът на 1-грамите в оценявания превод, които се срещат в референтния превод.
- P_2 е прецизността на 2-грамите
- P_3 е прецизността на 3-грамите
- P_4 е прецизността на 4-грамите
- $|\bar{\mathbf{y}}|$ е дължината на референтния превод
- $|\mathbf{y}|$ е дължината на оценявания превод

Обобщение

- Архитектурите “последователност към последователност” с внимание дават чудесни резултати върху задачи като машинен превод, резюмиране на документ, разпознаване на реч и много други.
- Могат да се прилагат различни варианти на архитектурата — брой слоеве на РНМ, двупосочен енкодер, допълнителни перцептрони на изхода и т.н.
- Допълването с архитектура за внимание значително подобрява резултатите.
- Вниманието може да бъде добавено както на изхода, така и на входа на PHM (early / late binding). Може и на двете места.
- Влагането на думите на входния език може да се реализира с конволюция на символи или да се използва кодиране до поддуми (виж предишните лекции).
- За генерирането на непознати думи при извеждането на изходна дума за <UNK> символа може да се използва модел за генератор на символи. Виж:
  Luong and Manning, 2016: Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models, https://arxiv.org/abs/1604.00788
- Кой вариант дава най-добри резултати зависи от конкретната задача, език, корпус и т.н.