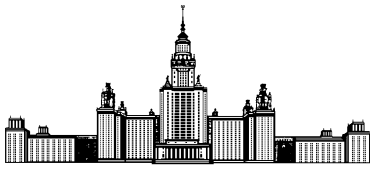
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Математические методы анализа текстов

Отчет по практическому заданию №3.

«Задача выравнивания в машинном переводе»

Выполнил:
магистр 1 курса 517 группы
Таскынов Ануар

Москва, 2018

Содержание.

1 Описание задачи

В данном задании необходимо реализовать, проанализировать и оценить несколько моделей для задачи выравнивания. Для обучения моделей был предоставлен англо-чешский корпус, по которому необходимо было EM-алгоритмом оценить сопоставления слов. За основу взята модель IBM Model 1.

2 Теория

Матрица сопоставления слов — это матрица A_{ij}, состоящая из нулей и единиц, причём $A_{ij} = 1$, если i–е слово из исходного предложения переводится в j–е слово переведенного.

Введём некоторые обозначения: e — исходное предложение, размер данного предложения будем обозначать через I; f — переведённое предложение, размер предложения J. Правдоподобие выборки можно записать следующим образом:

$$p(D|\theta) = \prod_{(f,e)\in D} P(f|e,\theta) = \prod_{(f,e)\in D} \sum_{a\in\mathcal{A}} p(a,f|e,\theta) = \prod_{(f,e)\in D} \sum_{a\in\mathcal{A}} p(a|e,\theta)p(f|e,a,\theta),$$

где D — корпус предложений, $\mathcal{A}$ — множество всех матриц сопостовлений слов ($|\mathcal{A}| = 2^{IJ}$), θ — параметры модели. Первое слагаемое в произведении $p(a|e,\theta)$ называют prior model, а $p(f|e,a,\theta)$ — translation model. Видно, что использовать такое сложение не выгодно, так что создатели IBM Model1 решили упростить модель:

1. каждому переведённому слову соответствует только одно слово из исходного предложения. Таким предположением количество элементов в сумме уменьшается с 2^{IJ} в I^J ($a_j \in [1, I]$).

2. каждое переведённое слово генерируется условно независимо от исходного слова, с которым оно выровнено. Таким образом:

$$p(f|e,\theta) = \sum_{a\in\mathcal{A}} p(a|e,\theta)p(f|e,a,\theta) \approx \prod_{j=1}^{J} \sum_{a^{\setminus j}\in\mathcal{A}} p(a^{\setminus j}|e,\theta) \sum_{a_j=1}^{I} p(a_j|a^{\setminus j},e,\theta)p(f_j|e_{a_j},\theta),$$

   где $a^{\setminus j}$ — вектор a без j–й позиции.

3. $p(a_j|a^{\setminus j},e,\theta) = p(a_j|e,\theta)$. Тогда:

$$p(f|e) = \prod_{j=1}^{J} \sum_{a_j=1}^{I} p(a_j|e,\theta)p(f_j|e_{a_j},\theta) \sum_{a^{\setminus j}\in\mathcal{A}} p(a^{\setminus j}|e,\theta) = \prod_{j=1}^{J} \sum_{a_j=1}^{I} p(a_j|e,\theta)p(f_j|e_{a_j},\theta)$$

4. Далее, $t(f_j|e_{a_j}) := p(f_j|a_j,e,\theta)$, а prior $p(a_j|e,\theta) = \epsilon$. Запишем нижнюю оценку на логарифм правдоподобия:

$$\mathbb{E}\Big[\log p(f,a|e)\Big] = \prod_{j=1}^{J} \sum_{a_j=1}^{I} p(a_j|f,e,\theta) \log p(f_j,a_j|e,\theta) = \prod_{j=1}^{J} \sum_{a_j=1}^{I} p(a_j|f,e,\theta) \log t(f_j|e_{a_j})\epsilon$$

Пересчёты для E и M-шагов в EM-алгоритме:

$$p(a_j = i|f,e,\theta) = \frac{p(a_j = i|e,\theta)p(f_j|a_j = i,e,\theta)}{\sum_{i'=1}^{I} p(a_j = i'|e,\theta)p(f_j|a_j = i',e,\theta)} = \frac{t(f_j|e_i)}{\sum_{i'=1}^{I} t(f_j|e_{i'})},$$

$$t(f|e)_{t+1} = \frac{\sum_{(\hat{f},\hat{e})\in D} \sum_j \sum_i p(a_j = i|f,e,\theta)\delta((\hat{f}_j = f)\&(\hat{e}_i = e))}{\sum_{(\hat{f},\hat{e})\in D} \sum_j \sum_i p(a_j = i|f,e,\theta)\delta(\hat{e}_i = e)}.$$

Данную модель можно улучшить с помощью добавления хоть какой-то дополнительной информации в prior, использовать POS-теги и так далее.

3 Модели выравнивания

В данной секции будут приведены результаты экспериментов и описание улучшенных моделей.

3.1 IBM Model1

Модель, которая была описана выше:

полнота: 0.509; точность: 0.529; aer: 0.481.

Можно улучшить качество, запустив данную модель на леммах (лемматизированные слова):

полнота: 0.526; точность: 0.546; aer: 0.464.

Также был дан корпус с 100000 предложениями (далее, просто 100K), на которых тоже протестируем модель:

полнота: 0.515; точность: 0.534; aer: 0.475

Данная модель на лемматизированных 100K предложениях:

полнота: 0.528; точность: 0.546; aer: 0.463

3.2 Модели, основанные на позициях слов

Можно добавить в prior, что переведённое слово и исходное находятся на одинаковых положениях относительно своих предложений. То есть сделать примерно следующее: $p(a_j = i|e, \theta) \sim \frac{1}{|\frac{i}{I} - \frac{j}{J}|+1}$.

Простая модель:

полнота: 0.588; точность: 0.611; aer: 0.401

Простая модель на леммах:

полнота: 0.606; Точность: 0.629; aer: 0.382

Простая модель на 100K:

полнота: 0.576; точность: 0.601; aer: 0.411

Простая модель на 100K с леммами:

полность: 0.595; точность: 0.620; aer: 0.393

Приведём также пример из pdf Дэвида Талбота: $p(a_j = i|e, \theta) = p(a_j = i|j, I, J)$. Тогда нижняя оценка на логариф правдоподобия будет следующей:

$$\mathbb{E}\Big[\log p(f, a|e)\Big] = \prod_{j=1}^{J}\sum_{a_j=1}^{I} p(a_j|f, e, \theta)\log p(f_j, a_j|e, \theta) = \prod_{j=1}^{J}\sum_{a_j=1}^{I} p(a_j|f, e, \theta)\log p(i|j, I, J)t(f_j|e_{a_j}).$$

Пересчёт для $t(f|e)$ остаётся таким же. Рассмотрим, как изменятся $p(i|j, I, J), p(a_j = i|f, e, \theta)$:

$$p(a_j = i|f, e, \theta) = \frac{p(a_j = i|e, \theta)p(f_j|a_j = i, e, \theta)}{\sum_{i'=1}^{I} p(a_j = i'|e, \theta)p(f_j|a_j = i', e, \theta)}$$

$$p(i|j, I, J) = \frac{\sum_{(f,e)\in D} p(a_j = i|f, e, \theta)\delta((|e| = I)(|f| = J))}{\sum_{(f,e)\in D}\sum_{i'=1}^{I} p(a_j = i|f, e, \theta)\delta((|e| = I)(|f| = J))}$$

Посмотрим на результаты качества работы модели:

полнота: 0.513; точность: 0.539; aer: 0.474

Данная модель на лемматизированных предложениях:

полнота: 0.534; точность: 0.559; aer: 0.454

Данная модель на 100K предложениях:

полнота: 0.588; точность: 0.625; aer: 0.393

Данная модель на лемматизированных предложениях на 100K:

полнота: 0.601; точность: 0.638; aer: 0.381

3.3 Модели, основанные на POS-тегах

Также можно оценить prior следующим образом: $p(a_j = i) = p(\text{tag}(f_j)|\text{tag}(e_i))$. Посмотрим на качество работы:

полнота: 0.530; точность: 0.551; aer: 0.459

Прайор на POS-тегах на 100K:

полнота: 0.527; точность: 0.547; aer: 0.463

Прайор на POS-тегах и на лемматизированных предложениях:

полнота: 0.544; точность: 0.564; aer: 0.446

Прайор на POS-тегах и на леммах 100K:

полнота: 0.536; точность: 0.556; aer: 0.454

Можно попробовать объединить данную идею с POS-тегами и позициями:

$$p(a_j = i|e, \theta) \sim \frac{p(\text{tag}(f_j)|\text{tag}(e_i))}{|\frac{i}{I} - \frac{j}{J}| + 1}.$$

Данная модель:

полнота: 0.604; точность: 0.627; aer: 0.384

Данная модель на 100K предложениях:

полнота: 0.587; точность: 0.613; aer: 0.400

Данная модель на лемматизированных предложениях:

полнота: 0.619; точность: 0.642; aer: 0.370

Данная модель на лемматизированных 100K предложениях:

полнота: 0.601; точность: 0.626; aer: 0.386

Можно также изменить translation model: $p(f_j|e_i) \approx p(f_j, \text{tag}(f_j)|e_i, \text{tag}(e_i))$ и использовать вместе с приведённым выше prior model.

Данная модель:

полнота: 0.595; точность: 0.622; aer: 0.391

Данная модель на лемматизированных предложениях:

полнота: 0.608; точность: 0.634; aer: 0.379

Модель не удалось запустить на 100K датасете (не хватило памяти).

4 Инструкции по запуску

Для запуска скрипта word_alignment нужно также добавить в командную строку аргумент tags для использования POS-тегов, иначе добавить любой символ. Все классы для приведённых выше моделей реализованы в виде классов PriorModel модуле models.py. Для того, чтобы переключать модели между собой необходимо в модуле word_alignment.py в функции initialize_models переключить prior (PriorModel4, PriorModel5, TranslationModel2 работают только с POS-тегами).

5 Выводы

В данной работе были проделаны несколько модификаций исходного алгоритма IBM Model1 на англо–чешском корпусе с 10000 и 100000 предложениями. Предложено 4 дополнительных способа задания prior. Также был реализован translation model на основе POS–тегов. Приведём лучшие модификации IBM Model1:

- На датасете 10K лучшее качество показал prior на основе POS–тегов и позиций (полнота: 0.604; точность: 0.627; aer: 0.384).
- На датасете 10K лемматизированных предложений также сработа prior на основе POS–тегов и позиций (полнота: 0.619; точность: 0.642; aer: 0.370).
- На датасете 100K предложений лучшее качество было у модели на основе позиций из конспекта Дэвида Талбота(полнота: 0.588; точность: 0.625; aer: 0.393).
- На датасете 100K лемматизированных предложений лучше сработал модель на основе позиций из pdf Дэвида Талбота(полнота: 0.601; точность: 0.638; aer: 0.381).

Prior'ы на основе позиций выигрывают, когда два языка такие, что исходное и переведённые слова будут в одинаковых положениях относительно предложений (это не всегда так). Если к этому еще добавить встречаемости тегов в предложениях, то эту модель можно усилить, как это было видно в экспериментах.

Пересчёт prior model на каждой итерации EM алгоритма переобучает саму модель (видимо, нужно prior model пересчитывать на отложенной выборке). Однако эта модель сработала лучше для 100K предложений (в том числе и лемматизированных).

Translation model брать в зависимости от POS–тегов также переобучает алгоритм (если смотреть на логарифм правдоподобия).