1 Wprowadzenie

Tłumaczenie między językami naturalnymi ma duże znaczenie praktyczne. Od dawna próbowano je zautomatyzować.

Problemy:

- wieloznaczne słowa
- zmiany w szyku zdania
- słowa funkcyjne i konstrukcje gramatyczne zmieniają znaczenie słów

Najprostszą metodą jest użycie słownika, każdemu słowu z języka źródłowego przypisując jego odpowiednik ze słownia. Przykłady:

Boy goes home

może dać

Chłopiec idzie dom

Jest to sensowne, ale brzmi niezbyt dobrze.

Duch jest dobry

może dać

Spirit is good

Co z kolei może dać

Spirytus jest dobry

Pokazuje to że wieloznaczność słów może łatwo prowadzić do przekłamań.

Przez wiele lat panowało przekonanie że by poprawić jakość tłumaczenia należy przeprowadzić analizę gramatyczną (rozbiór) zdania źródłowego. Proponowano użycie przypadków dla ograniczenia wieloznaczności, np. informacje gramatyczne pozwalają stwierdzić że właściwym tłumaczniem w pierwszym przykładzie jest

Chłopiec idzie do domu

Niestety, okazało się to trudne. W praktyce lepsze wyniki osiągano stosując metody ad hoc. Dużym problem była ilość pracy potrzebna do stoworzenia systemu tłumaczenia, zarówno metody ad hoc jak i bardziej systematyczne podejścia wymagały starannego ręcznego kodowania dużych ilości informacji.

W początkach tłumaczenia maszynowego próbowano użyć podejście statystyczne. Jednak pierwsze próby nie dały zadawalających wyników i przez wiele lat panowało przekonanie że metody statystyczne są nieprzydatne do

tłumacznia. Zmieniło się to po roku 1980. Wtedy zespół z IBM zaproponował statystyczną metodę tłumacznia. Podstawą był tzw. tekst równoległy, to znaczy długi tekst z odpowiadającym mu tłumaczeniem. Zespół z IBM używał Hansard, tzn. protokóły posiedzeń parlamentu Kanadyjskiego. W Kanadzie są dwa języki urzędowe, angielski i francuski. Wystąpienia w jednym języku są tłumaczone na drugi, tak że w efekcie dostajemy tekst równoległy w języku angielskim i francuskim. Nowsze prace w Europie używały protokóły Parlamentu Europejskiego które są tłumaczone na wszystkie jezyki oficjalne Uni Europejskiej.

Istotną trudnością dla metod statystycznych jest rzadkość danych. Konsekwencją prawa Zipfa jest to że większość słów występuje rzadko. Poniżej jako ilustrację pokazuję analizę dziennika ustaw z 2013 roku. Dane wejściowe to były publicznie dostępne pliki PDF. Skonwertowałem je do tekstu i utworzyłem listę słów wraz z częstościami wystąpień. Ze względu na proces konwersji jest sporo śmieci, jednak jakościowo wyniki dla lepszych tekstów (również dla innych języków) są podobne. Języki bez fleksji przy danej długości tekstu będą miały mniej różnych słów, jednak w typowym tekście słowa występujące jednokrotnie stanowią sporą część wszystkich słów.

Kategoria	licznik	suma
do 1	704907	704907
do 2	265986	531972
do 5	205125	756315
do 10	99368	749162
do 18	54597	760006
do 30	33055	783996
do 50	23791	932359
do 100	22108	1567511
do 250	18075	2835379
do 600	9664	3689787
do 1500	5023	4605421
do 10000	3171	10618403
do 100000	455	10634595
do 1000000	39	9466664
do 10000000	1	1225113
Razem	1445365	49861590

Widać też że jest kompensujący czynnik: cały tekst ma ponad 49 milionów słów, słów występujących jednokrotnie jest 704907 czyli ok. 1.4 procent.

Zwykle o tekście równoległym zakłada się że jest on podzielny na zdania, tak że odpowiadające sobie zdania są swoimi tłumaczeniami. Zwykle surowe teksty źródłowe nie są tak podzielone, ale wyodrębnianie zdań nie jest

zbyt trudne, a odpowiedniość między zdaniami można wyznaczyć metodami statystycznymi.

Najnowsze metody tłumaczenia używają sieci neuronowe (uczymy je na tekście równoległym), jednak dalej powiemy o metodach statystycznych.

2 Model zaszumionego kanału

Modele IBM traktują tłumaczenia jako problem odtwarznia sygnału który przesłano przez zaszumiony kanał transmisyjny: tekst w naszym języku N jest po przejsciu przez kanał transmisyjny zniekształcony i wychodzi jako takst w języku obcym F.

Aby uzyskać z powrotem tekst N maksymalizujemy prawdopodobińśtwo warunkowe $P(N|F)$, czyli obliczamy

$$\operatorname*{argmax}_N P(N|F)$$

Mamy

$$P(N|F) = \frac{P(N \cap F)}{P(F)} = \frac{P(N \cap F)P(N)}{P(N)P(F)} = \frac{P(F|N)P(N)}{P(F)}$$

czyli

$$\operatorname*{argmax}_N P(N|F) = \operatorname*{argmax}_N \frac{P(F|N)P(N)}{P(F)} = \operatorname*{argmax}_N P(F|N)P(N)$$

gdzie ostatnia równość zachodzi bo $P(F)$ jest niezależne od N i nie wpływa na to gdzie jest osiągane maksimum.

Powyższe przekształcenie jest pomocne, bo przy bezpośrednim użyciu $P(N|F)$ potrzebujemy bardzo dobre oszacownie prawdopodobieństwa. W $P(F|N)P(N)$ mamy rozdzielne różne zadania: $P(F|N)$ dba o to by F i N dobrze sobie odpowiadały, ale nie musi się troszczyć o to czy N i F są dobrze zbudowanie. O jakość N troszczy się $P(N)$, tu używamy danych tylko dla jednego języka co jest łatwiej dostępne niż teksty równoległe. Zalety tego łatwo widać w przypadku Modelu 1 IBM: $P(F|N)$ jest nieczułe na zmiany kolejności słów, ale $P(N)$ preferuje zdania z właściwą kolejnością

Model zaszumionego kanału pojawił się w zagadnieniach technicznych. Przy przewarzaniu języka był wcześniej użyty do rozpoznawania mowy, model tłumacznia zaadoptował i rozwinął techniki stosowane wcześniej dla mowy.

3 Model języka

W statystycznym tłumaczeniu maszynowym typowo używa się modele Markowa na słowach. Klasyczny model Markowa na zbiorze W każdej parze w, v elementów W przypisuje prawdopobieństwo przejścia $p_{w,v}$ tak że

$$P(w_n = w|w_1 w_2 \dots w_{n-1}) = P(w_n = w|w_{n-1}) = p_{w,w_{n-1}}).$$

W zagadnieniach języka taki model często jest za słaby bo modeluje tylko zależności między kolejnymi słowami.

W praktyce używa się modele Markowa wyższego rzędu. Teoretycznie Markowa rzędu k można zdefinowiać jako model Markowa na W^k, taki że

$$p_{w_1 w_2 \dots w_k, v_1 v_2 \dots v_k}$$

jest różne od zera tylko wtedy gdy dla $i = 1, 2, \dots, k-1$ zachodzi $w_i = v_{i+1}$. Innymi słowy,

$$P(w_n = w|w_1 w_2 \dots w_{n-1}) = P(w_n = w|w_{n-k} w_{n-k+1} \dots w_{n-1}) = p_{w,w_{n-k} w_{n-k+1} \dots w_{n-1}}$$

i trzeba tylko podać $p_{w,v}$ gdzie w to słowo zaś v to ciąg słów długości k. W praktyce k często jest większe lub równe 4, a więc chcemy szacować częstości wystąpień ciągów słów długości co najmniej 5. Typowy język zawiera więcej niż $100000 = 10^5$ słów, toteż dla rzędu 4 trzeba $(10^5)^5 = 10^{25}$ prawdopodobieństw. Jasne jest że dla tekstów praktycznych rozmiarów nie da się oszacować wszystkich prawdopodobieństw przez częstości wystąpień, większość ciągów będzie miała częstość zero. Używa się tu estymację Bayesa. Dla większych k często trafiamy na ciągi których nie było w zbiorze treningowym i dlatego mamy tylko szacowanie Bayesowskie. Ale w takim przypadku któtszy ciąg może być w zbiorze treningowym i ma sens użyć pochodzące stąd oszacowanie. Praktycznie realizuje się to przez kombinację wypukłą modeli różnych rzędów:

$$q_{w,v} = \sum_{i=1}^{k} c_i p_{w,v_i v_{i+1} \dots v_k}$$

gdzie c_i to waga modelu rzędu $k + 1 - i$.

Komentarz praktyczny: prawdopodobieństwa mogą być bardzo małe, tak że w arytmetyce komputerowej są nieodróżnialne od zera. Dlatego zwykle pracuje się z logarytmami. Logarytmy upraszczają mnożenie zaś komplikują dodawanie. Na współczesynch komputerach oznacza to że obliczenia na logarytmach będą bardziej kosztowne niż bezpośrednie operacje na prawdopodobieństwach. Ale koszt logarytmów zwykle jest mniejszy niż koszt wielokrotnej precyzji potrzebnej by uzyskać odpowiedni zakres liczb. Bardziej pracochłonną, ale obliczniowo szybszą alternatywą może być skalowanie.

4 Modele IBM

Badacze IBM zdefiniowali ciąg coraz dokładniejszych modeli, Model 1, Model 2 aż do Model 5.

- Model 1 zakłada że wszystkie dopasowania są jednakowo prawdopodobne, tak że porządek słów nie odgrywa roli
- Model 2 zakłada że prawdopodobieństwo dopasowania zależy od pozycji słów i długości zdań, ale nie zależy od słów
- Model 3 uwzględnia że jedno słowo może mieć więcej tłumaczeń (płenność, ang. fertility)
- Model 4 uzależnia prawdopodobieństwo dopasowania od odległości między słowami
- Model 5 poprawia Model 4 usuwając część bzdurnych odpowiedniości.

Dlaczego tyle modeli? Czy nie było by lepiej używać tylko najbardziej zaawansowany (Model 5)? Bardziej zaawansowane modele są trudne obliczniowo. Model 1 używa się do inicjowania Modelu 2, ten do Modelu 3 itd. Nie są znane efektywne metody bezpośredniego uczenia wyższych modeli IBM, dlatego są potrzebne wszyskie.

W modelach IBM istotną rolę odgrywa tzn. dopasowanie, tzn. relacja mówiąca które słowa odpowiadają sobie w tłumaczeniu. Np. w parze

Boy goes home

i

Chłopiec idzie do domu

słowa *do* oraz *domu* odpowiadają słowu *home*. Pokazuje to że jedno słowo może mieć więcej odpowiedników. Jedną z istotnych różnic między modelami jest to jakie dopasowania są dopuszczalne i jakie mają one prawdopodobieństwa.

5 Model 1

W modelu IBM 1 zakładamy że słowa zdania w języku obcym F są tłumaczone niezależnie od siebie. Każde z nich jest tłumaczeniem pewnego słowa ze zdania w naszym języku, przy tym to które słowo z naszego języka odpowiadające słowu j-temu słowu zdania F (które oznaczymy przez F_j) jest opisane przez rozkład jednostajny. Dokładniej, oznaczmy przez l_N długość zdania N w naszym języku a przez l_F długość zdania w języku obcym.

Aby dopuścić słowa w języku obcym które nie mają odpowiednika w naszym języku przed N dodajemy fikcyjne słowo na pozycji 0. Odpowiedniość (dopasowanie) między słowami w zdaniu F i zdaniu N jest zadana przez funkcję $a : [1, 2, \ldots, l_F] \to [0, 1, \ldots, l_N]$, tzn. słowo F_j jest tłumaczeniem $N_{a(j)}$. Wynika stąd że niektóre słowa z N mogą nie mieć odpowiednika przy tłumaczniu (efektywnie znikać).

Parametrami modelu są prawdopodobieństwo tłumaczenia słów, tzn. mamy tabelę pozwalającą obliczyć $P(F_j|N_i)$. Dla funkcji a przyjmujemy rozkład jednostajny, czyli dla każdego j wartość $a(j)$ ma rozkład jednostajny i dla różnych j rozkłady są niezależne.

Łącznie prawdopodobieństwo zdania w języku obcym F o długości l_F i dopasowania a pod warunkiem zdania w naszym języku N o długości l_N wynosi:

$$P(F, a|N) = \frac{\epsilon}{(1 + l_N)^{l_F}} \prod_{j=1}^{l_F} P(F_j|N_{a(j)})$$

gdzie ϵ (zależne od l_F i l_N, ale niezależne od a i F) jest dobrane tak by prawdopodobieństwa sumowały się do 1.

Mamy

$$P(F|N) = \sum_a P(F, a|N) = \frac{\epsilon}{(1 + l_N)^{l_F}} \sum_{a(1)=0}^{l_N} \cdots \sum_{a(l_F)=0}^{l_N} \prod_{j=1}^{l_F} P(F_j|N_{a(j)})$$

$$= \frac{\epsilon}{(1 + l_N)^{l_F}} \prod_{j=1}^{l_F} \sum_{a(j)=0}^{l_N} P(F_j|N_{a(j)})$$

$$= \frac{\epsilon}{(1 + l_N)^{l_F}} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i).$$

Powyższe przekształcenie wygląda niewinnie, ale jest to dramatyczna poprawa efektywności obliczeń: zastępujemy sumę wykładniczo wielu (względem długości l_F) produktów produktem sum mającym łącznie $l_F(L_N + 1)$ wyrazów.

Do wyznaczania parametrów model IBM 1 maksymalizuje

$$\prod_{(F,N)\in T} P(F|N)$$

gdzie T to zbiór par użyty do uczenia. Przechodząc to logarytmów jest to równoważne maksymalizacji

$$L = \sum_{(F,N)\in T} \log(P(F|N)).$$

Mamy

$$P(F|N) = c_{F,N} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i)$$

co prowadzi do równości

$$\log(P(F|N)) = \log(c_{F,N}) + \sum_{j=1}^{l_F} \log(\sum_{i=0}^{l_N} P(F_j|N_i))$$

i łącznie

$$L = c + \sum_{(F,N)\in T} \sum_{j=1}^{l_F} \log(\sum_{i=0}^{l_N} P(F_j|N_i)).$$

Jako że logarytm jest funkcją ściśle wklęsłą L jest funkcją wklęsłą od parametrów $P(F_j|N_i)$ i ma jednoznaczną wartość maksymalną. Innym słowy, istnieją jednoznacznie wyznaczone liczby $\lambda_{F,j}$ takie że

$$\max L = c + \sum_{(F,N)\in T} \sum_{j=1}^{l_F} \lambda_{F,j}$$

i

$$\log(\sum_{i=0}^{l_N} P(F_j|N_i)) = \lambda_{F,j}.$$

A więc

$$\sum_{i=0}^{l_N} P(F_j|N_i) = \exp(\lambda_{F,j}).$$

Razem z warunkami $\sum_w P(w|N_i) = 1$ i $P(F_j|N_i) \geq 0$ daje to jednoznacznie wyznaczony układ równań i nierówności liniowych na wartości $P(F_j|N_i)$ dające maksimum L. W literaturze były błędne stwierdzenia że powyższy układ równań i nierówności ma jednoznaczne rozwiązanie. Niestety, nie jest to prawdą, rozwiązanie nie musi być jednoznaczne. Niejednoznaczność łatwo sztucznie wyprodukować: jeśli każde wystąpienie słowa F_j zastąpimy przez parę słow F_k i F_l, to w wyrażeniach wyżej $P(F_j|N_i)$ zostanie zastąpione przez

$$P(F_k|N_i) + P(F_l|N_i)$$

czyli jedyne ograniczenia na $P(F_k|N_i)$ i $P(F_j|N_i)$ to warunek nieujemności i to by w sumie dały stare $P(F_j|N_i)$. Jeśli stare $P(F_j|N_i) > 0$, to mamy nieskończenie wiele rozwiązań.

6 Algorytm EM

Bezpośrednie szacowanie $P(F_j|N_i)$ przez maksymalizaję prawdopodobieństwa zbioru treningowego jest kłopotliwe i niepotrzebne. Jeśli znamy dopasowanie to można użyć częstościowe oszacowanie $P(F_j|N_i)$ po prostu zliczając ile razy słowo F_j odpowiada N_i i dzieląc to przez ilość słów odpowiadających N_i. Jeśli znamy tylko prawdopodobieństwa dopasowań, to liczymy wartość oczekiwaną względem dopasowania. Tzn.

$$c(f,n) = \sum_{(F,N)\in T} \sum_{a} P(a|F,N)c(a,f,F,n,N)$$

gdzie $c(f,n)$ to oczekiwana ilość odpowiedniości między słowami f i n, zaś $c(a,f,F,n,N)$ to ilość odpowiedniości między f i n w zdaniach F i N przy dopasowaniu a.

Z wzoru Bayesa

$$P(a|F,N) = P(F,a|N)/P(F|N).$$

$P(F,a|N)$ jest zadane przez definicję Modelu 1:

$$P(F,a|N) = \frac{\epsilon}{(1+l_N)^{l_F}} \prod_{j=1}^{l_F} P(F_j|N_{a(j)})$$

$P(F|N)$ wyliczyliśmy wcześniej

$$P(F|N) = \frac{\epsilon}{(1+l_N)^{l_F}} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i)$$

czyli

$$\begin{aligned} P(a|F,N) &= P(F,a|N)/P(F|N) \\ &= \frac{\frac{\epsilon}{(1+l_N)^{l_F}} \prod_{j=1}^{l_F} P(F_j|N_{a(j)})}{\frac{\epsilon}{(1+l_N)^{l_F}} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i)} \\ &= \frac{\prod_{j=1}^{l_F} P(F_j|N_{a(j)})}{\prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i)} \\ &= \prod_{j=1}^{l_F} \frac{P(F_j|N_{a(j)})}{\sum_{i=0}^{l_N} P(F_j|N_i)} \end{aligned}$$

Teraz można lepiej obliczać $c(f,n)$. Mianowicie

$$c(f,n) = \sum_{(F,N)\in T} \sum_{a} P(a|F,N)c(a,f,F,n,N)$$

i

$$c(a,f,F,n,N) = \sum_{j=1}^{l_F} \delta(f,F_j)\delta(n,N_{a(j)})$$

Następnie

$$\sum_{a} P(a|F,N)c(a,f,F,n,N) = \sum_{a} P(a|F,N) \sum_{j=1}^{l_F} \delta(f,F_j)\delta(n,N_{a(j)})$$

$$= \sum_{j=1}^{l_F} \delta(f,F_j) \sum_{a} P(a|F,N)\delta(n,N_{a(j)})$$

Dalej, przy ustalonym j używając wzór na $P(a|F,N)$

$$\sum_{a} P(a|F,N)\delta(n,N_{a(j)})$$

$$= \sum_{a(1)=0}^{l_N} \cdots \sum_{a(l_F)=0}^{l_N} \delta(n,N_{a(j)}) \prod_{k=1}^{l_F} \frac{P(F_k|N_{a(k)})}{\sum_{i=0}^{l_N} P(F_k|N_i)}$$

Dla $k \neq j$ mamy

$$\sum_{a(k)=0}^{l_N} \frac{P(F_k|N_{a(k)})}{\sum_{i=0}^{l_N} P(F_k|N_i)} = \frac{\sum_{i=0}^{l_N} P(F_k|N_i)}{\sum_{i=0}^{l_N} P(F_k|N_i)} = 1.$$

Czyli (dalej przy ustalonym j)

$$\sum_{a} P(a|F,N)\delta(n,N_{a(j)}) = \sum_{a(j)=0}^{l_N} \delta(n,N_{a(j)}) \frac{P(F_j|N_{a(j)})}{\sum_{i=0}^{l_N} P(F_j|N_i)}$$

$$= \frac{\sum_{i=0}^{l_N} \delta(n,N_i)P(F_j|N_i)}{\sum_{i=0}^{l_N} P(F_j|N_i)} = \frac{\sum_{i=0}^{l_N} \delta(n,N_i)P(F_j|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)}$$

$$= \frac{P(F_j|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{i=0}^{l_N} \delta(n,N_i)$$

Wstawiając to do poprzednich wzorów mamy

$$c(f,n) = \sum_{(F,N)\in T} \sum_{j=1}^{l_F} \delta(f,F_j) \frac{P(F_j|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{i=0}^{l_N} \delta(n,N_i)$$

$$= \sum_{(F,N)\in T} \frac{P(f|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{j=1}^{l_F} \delta(f,F_j)\delta(n,N_i)$$

Teraz można użyć $c(f,n)$ do oszacownia $P(f|n)$

$$P(f|n) = \frac{c(f,n)}{\sum_f c(f,n)}$$

Zauważmy jeszcze że sumę z mianownika można uprościć:

$$\sum_f c(f,n) = \sum_f \sum_{(F,N)\in T} \frac{P(f|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{j=1}^{l_F} \delta(f,F_j) \sum_{i=0}^{l_N} \delta(n,N_i)$$

$$= \sum_f \sum_{(F,N)\in T} \frac{1}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{j=1}^{l_F} \delta(f,F_j) P(F_j|n) \sum_{i=0}^{l_N} \delta(n,N_i)$$

$$= \sum_{(F,N)\in T} \frac{\sum_{j=1}^{l_F} P(F_j|n)}{\sum_{i=0}^{l_N} P(F_j|N_i)} \sum_{i=0}^{l_N} \delta(n,N_i)$$

Są to stosunkowo wydajne wzory, wymagają rzędu

$$t = \sum_{(F,N)\in T} l_F(1+l_N)$$

operacji. Przy rozsądnym założeniu że średnie $l_F(1+l_N)$ jest ograniczone oznacza to liniowy wzrost względem rozmniaru zbioru treningowego.

Zaczynając z jednostajnego oszacowania $P(f|n)$ na parach słów które pojawiają się się w którejś z par zdań zbioru treningowego (takich par jest t) po kilku iteracjach dostaniemy rozsądne oszacowanie $P(f|n)$. W praktyce nie należy używać zbyt wielu iteracji. Mianowicie, nasze oszacownie chcemy użyć jako wartość początkową dla Modelu 2. Zbyt duża ilość iteracji w Modelu 1 prowadzi do przetrenowania.

Komentarz: naukę Modelu 1 zaczynamy od równych prawdopodobieństw. Jeśli nie ma jednosnacznego rozwiązania, to nasze inicjowanie automatycznie wybiera jedno z rozwiązań (najbardziej symetryczne rozwiązanie).

7 Ogólniej o algorytmie EM

To co robiliśmy poprzednio może wyglądać dość ad-hoc. Ale jest to faktycznie szczególny przypadek ogólnego algorytmu EM.

Niech będzie dany zestaw obserwacji x i sparametryzowna rodzina rozkładów prawdopodobieństwa P_θ z gęstością p_θ. Metoda estymacji największej wiarygodności polega na maksymalizacji $p_\theta(x)$, tzn. szukamy θ dla którego $p_\theta(X)$ jest maksymalne. Jednakże bezpośrednia maksymalizacja $p_\theta(X)$ może być bardzo trudna. Algorytm EM jest iteracyjną procedurą która może być łatwiejsza od bezpośredniej maksymalizacji.

Poniżej będziemy zakładać że wszystkie potrzebne nam rozkłady prawdopodobieństwa mają gęstości, co w szczególności jest spełnione w przypadku dyskretnej przestrzeni probabilistycznej (choć dla uproszczenia notacji poniżej używamy symbole tak jakbyśmy mieli przypadek ciągły).

W algorytmie EM zakładamy że istnieje nieobserwowalna wielkość y która w pewnym sensie wyznacza x. Dokładniej, zakładamy że rozkład warunkowy (a właściwie gęstość warunkowa) x pod warunkiem y nie zależy od θ czyli oznaczając przez r_θ gęstość y, przez $q(x|y)$ gęstość warunkową x pod warunkiem y, zaś przez q_θ gęstość łączną mamy

$$q_\theta(x, y) = r_\theta(y)q(x|y).$$

W Modelu 1 maksymalizowaliśmy

$$\prod_{(F,N)\in T} P(F|N)$$

gdzie T to zbiór par użyty do uczenia. Użycie prawdopobieństwa warunkowego nie zmienia istoty rzeczy. Nasze x to zbiór trenigowy T. Parametrami iθ jest zbiór prawdopodobieństw warunkowych $P(f|n)$. y składa się ze zbioru treningowego T i dopasowań dla każdej pary (F, N). Ponieważ x jest jednoznacznie wyznaczone przez y to oczywiście $q(x|y)$ nie zależy od θ ($q(x|y) = 1$ jeśli x jest częścią y, 0 w przeciwnyme razie).

Komentarz: jest to bardzo częsta sytuacja gdy używamy algorytm EM, ale powyższy warunek jest słabszy a bazowane na nim rozumowanie jest co najmniej tak przejrzyste jak szczególny przypadek, więc wybieramy bardziej ogólną wersję.

Poniżej będziemy zakładać że zbiór y takich że $r_\theta(y) > 0$ nie zależy od θ. Z tego założenia wynika że zbiór par (x, y) takich że $q_\theta(x, y) > 0$ nie zależy od θ.

W algorytmie EM budujemy ciąg coraz lepszych przybliżeń θ_n do optymalnego θ. Poniżej wyprowadzimy kilka nierówności pokazujacych własności alogrytmu EM.

Niech $q_\theta(y|x)$ będzie gęstością warunkową y pod warunkiem x. Mamy

$$q_\theta(x,y) = p_\theta(x)q_\theta(y|x).$$

Teraz, zakładając że $q_\theta(x,y) > 0$ (czyli również $q_{\theta_n}(x,y) > 0$) mamy $q_{\theta_n}(y|x) > 0$, $q(x|y) > 0$, $p_{\theta_n}(x) > 0$ i

$$\frac{q_\theta(x,y)}{p_{\theta_n}(x)} = \frac{q_\theta(x,y)q_{\theta_n}(y|x)}{p_{\theta_n}(x)q_{\theta_n}(y|x)} = \frac{q_\theta(x,y)}{q_{\theta_n}(x,y)}q_{\theta_n}(y|x) =$$

$$\frac{r_\theta(y)q(x|y)}{r_{\theta_n}(y)q(x|y)}q_{\theta_n}(y|x) = \frac{r_\theta(y)}{r_{\theta_n}(y)}q_{\theta_n}(y|x)$$

Z powyższego wynika że dla $p_\theta(x) > 0$ mamy

$$\frac{p_\theta(x)}{p_{\theta_n}(x)} = \int \frac{q_\theta(x,y)}{p_{\theta_n}(x)}dy = \int_{q(x,y)>0} \frac{q_\theta(x,y)}{p_{\theta_n}(x)}dy = \int \frac{r_\theta(y)}{r_{\theta_n}(y)}q_{\theta_n}(y|x)dy.$$

W szczególności z istnienia (x,y) takiego że $q_\theta(x,y) > 0$ wynika że $p_{\theta_n}(x) > 0$.

Zauważmy teraz że logarytm jest funkcją ściśle wklęsłą. Przy ustalonym x oznaczając $L(\theta) = \log(p_\theta(x))$ z nierówności Jensena mamy

$$L(\theta) - L(\theta_n) = \log(\frac{p_\theta(x)}{p_{\theta_n}(x)}) = \log(\int \frac{r_\theta(y)}{r_{\theta_n}(y)}q_{\theta_n}(y|x)dy)$$

$$\geq \int \log(\frac{r_\theta(y)}{r_{\theta_n}(y)})q_{\theta_n}(y|x)dy = Q(\theta,\theta_n).$$

gdzie ostatnia równość jest definicją $Q(\theta,\theta_n)$. W algorytmie EM wybieramy jako θ_{n+1} takie θ które zmaksymalizuje $Q(\theta,\theta_n)$, albo przynajmniej takie by $Q(\theta_{n+1},\theta_n) > 0$. Przy takim wyborze θ_n wartości $L(\theta_n)$ tworzą ciąg ściśle rosnący. Jeśli dla dowolnego ustalonego x gestość $p_\theta(x)$ jest ograniczona, to również $L(\theta)$ jest ograniczona i ciąg $L(\theta_n)$ jest ograniczony, a wiec jest ciągiem zbieżnym. Przy rozsądnych założeniach, np. że dla dowolnego a zbiór θ takich że $L(\theta) \geq a$ jest zwarty, zaś Q ma ciągłą pochodną wynika stąd istnienie podciągu ciągu θ_n zbiegającego do θ_∞ takiego że $Q(\theta_\infty,\theta_\infty) = 0$ daje punkt stacjonarny $Q(\theta,\theta_\infty)$ (tzn. $\nabla_\theta Q(\theta,\theta_\infty) = 0$).

Zauważmy że $Q(\theta,\theta) = L(\theta) - L(\theta) = 0$, czyli jeśli $Q(\theta,\theta_n)$ nie osiąga maksimum dla $\theta = \theta_n$ to istnieje θ takie że $Q(\theta,\theta_n) > 0$. Jak zauważyliśmy wyżej przy słabych założeniach o Q istnieje podciąg zbieżny do θ_n i pochodna $\nabla_\theta Q(\theta,\theta_n) = 0$. Czyli jeśli również L jest różniczkowalna to pochodna L w θ_∞ jest równa 0. Czyli θ_∞ jest punktem stacjonarnym L. W specjalnych przypadkach może się zdarzyć że θ_∞ to punkt siodłowy, ale zwykle jest to maksimum lokalne. Ogólnie punkt stacjonarny może nawet być minimum

lokalnym, jednak dla $\theta_n \to \theta_\infty$ jest to wykluczone bo $L(\theta_n)$ tworzą ciąg ściśle rosnący. Model 1 jest wypukły, więc punkt stacjonarny jest maksimum globalnym. Przy brzegu funkcja $L(\theta)$ dąży do $-\infty$, a więc θ_n pozostają w zbiorze zwartym i mamy globalną zbieżność.

Dla Modelu 1 $q_{\theta_n}(y|x)$ to prawdopodobieństwa dopasowań przy ustalonych parametrach, czyli całka w definicji $Q(\theta, \theta_\infty)$ sprowadza się do sumy po dopasowaniach. Nasza procedura zliczania dająca nowe parametry maksymalizuje $Q(\theta, \theta_\infty)$. Dowód tego to wariant klasycznego twierdzenia że dla tzw. rodzin wykładniczych średnia z próby jest estymatorem największej wiarygodności.

8 Model 2

W Modelu 1 zakładaliśmy że j-te słowo F dopasowuje się z rozkładem jednostajnym ze słowami z N. W Modelu 2 zakładamy ustalony rozkład zależny od j i długości słów (ale nie zależący od samych słów). Wzory z Modelu 1 nieco się komplikują, ale dalej są wydajne. Model 2 nie jest wypukły, więc nie ma gwarancji zbieżności do optimum, ale inicjowanie wynikiem z Modelu 1 zwykle daje dobry resultat.

9 Model 3 i wyższe

W Modelu 3 każde słowo n w naszym języku ma przypisaną plenność (ang. fertility) ϕ_n. Plenność zależy tylko od słowa n zaś nie zależy od innych słów w zdaniu. Słowo n daje ϕ_n słów w języku obcym. Słowa w języku obcym są przestawione, każdemu przypisuje się pozycję. Pozycje słów w języku obcym zależą tylko od pozycji odpowiadającego mu słowa n. Popatrzmy na angielskie zdanie

For fertility zero , you drop the word

Jeśli językiem naszym jest angielski a chcemy z niego otrzymać polski (jako obcy) to naturalne jest przyjęcie że *you*, *the* oraz przecinek (który też traktujemy jako słowo) mają plenność zero, zaś pozostałe słowa mają plenność 1. Ponadto naturalne jest pozostawienie kolejności słów bez zmiany. Wybierając rozsądne odpowiedniki dla słów otrzymamy

Przy plenności zero pomijamy słowo

Gdy naszym językiem jest polski musimy użyć fikcyjne słowo na początku (tak jak w Modelu 1). To słowo będzie miało plenność 3 i wygeneruje słowa które znikały przy tłumaczeniu. Alternatywnie (z większym prawdopodebieństwem), słowo *pomijamy* otrzyma plenność 2 i wygeneruje frazę *we drop*,

zaś fikcyjne słowo też będzie miało plenność 2 i wygeneruje *the* oraz przecinek. W Modelu 3 plenność słowa fikcyjnego jest traktowana specjalnie. Najpierw wybieramy plenności normalnych słów co daje ilość odpowiadających im słów w języku obcym. Następnie dla każdego słowa tak wygenerowanego słowa w języku obcym generujemy lub nie dodatkowe słowa odpowadające słowu fikcyjnemu. Decyzja o tym czy generujemy dodatkowe słowo zależy od pojednyczego parametru p_0 zadającego prawdopodobieństwo generacji (w każdym kroku generujemy niezależnie). Dodatkowe słowa rozmieszczamy z równomiernym prawdopodobieństwem w wolnych pozycjach (nie zajętych przez normalne słowa).

Generację zdania w Modelu 3 można podsumować następująco:

1. dla każdego słowa ze zdania N wybieramy jego plenność z prawdopodobieństwem zależnym tylko od słowa
2. ustalamy ilość dodatkowych słów
3. z poprzednich kroków otrzymujemy długość zdania w języku obcym
4. dla każdego słowa w języku naszym generujemy opowiadające mu słowa w języku obcym, niezależnie z prawdopodobieństwem $P(f|n)$
5. dla każdego słowa normalnego słowa w języku obcym wybieramy jego pozcję z prawdopodobieństwem zależnym tylko od długości zdania, pozycji słowa źródłowego i numeru słowa w języku obcym
6. dodatkowe słowa rozmieszczamy jednostajnie w wolnych pozycjach

Parametrami Modelu 3 są prawdopodobieństwa plenności, prawdopodobieństwa słów $P(f|n)$, prawdopodobieństwa pozycji i prawdopodobieństwo p_0 generacji dodatkowych słów. Idea wyznaczania parametrów jest podobna jak dla Modelu 1: mając dopasowanie można oszacować parametry, mając parametry można obliczyć prawdopodobieństwa dopasowań. A więc dalej stosuje się algorytm EM. Obliczeniowo ten model jest bardziej skomplikowany bo nie działają uproszczenia z Modelu 1 i Modelu 2, tzn. nie nie daje się uprościć sumowania po dopasowanich i trzeba szacować sumy przy pomocy dopasowań o dużym prawdopodobieństwie.

Model 4 w bardziej skomplikowany sposób wybiera pozycję słów w języku obcym. Zarówno Model 3 jak i Model 4 mogą odwzorować dwa słowa w jedną pozycję. Przy tłumaczeniu takie odpowiedniości się nie pojawią i można ignorować ten problem. Ale można go wyeliminować pozwalając tylko na wybór wolnych pozycji, robi to Model 5. Nie jest jest jasne czy prowadzi to do praktycznej poprawy.

10 Ulepszenia

Modele IBM można bezpośrednio stosować do tłumaczenia, ale pojawiły się udoskonalenia. Mianowicie, szacując większość parametrów jak w modelach IBM rozszerzono tłumaczenie na frazy. Dało to najlepsze obecnie dostępne modele statystyczne.

Dokładniej, dopasowania na poziomie słów są istotne dla uczenia lepszych modeli (jak też dla innych zagadnień). Modele IBM czy ich warianty produkują dobre dopasowania.

Sporo uwagi poświęcono wariantom modeli IBM, w szczególności pojawiły się ulepszenia przy szukaniu dopasowań (lepsze modele dopasowania). Są też wypukłe warianty nieco lepsze niż Model 1. Podano też wariant Modelu 1 faktycznie mający jednoznaczne rozwiązanie.