Text Mining

Marcelo Mendoza
http://www.inf.utfsm.cl/~mmendoza
mmendoza@inf.utfsm.cl
A 131, Campus San Joaquín - UTFSM

N-grams

Texto como secuencia de tokens.

Sentencia: ‘its water is so transparent that’

¿Cuán probable es que la siguiente palabra sea ‘the’?

N-grams

Texto como secuencia de tokens.

Sentencia: ‘its water is so transparent that’

¿Cuán probable es que la siguiente palabra sea ‘the’?

Naive (necesita muchas observaciones para comparar pares):

N-grams

Texto como secuencia de tokens.

Sentencia: ‘its water is so transparent that’

¿Cuán probable es que la siguiente palabra sea ‘the’?

Naive (necesita muchas observaciones para comparar pares):

$$P(\textit{the}|\textit{its water is so transparent that}) = \frac{C(\textit{its water is so transparent that the})}{C(\textit{its water is so transparent that})}$$

Usando regla de la cadena de probabilidades (texto como secuencia):

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2})\ldots P(w_n|w_{1:n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_{1:k-1}) \end{aligned}$$

N-grams

Modelo bi-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

N-grams

Modelo bi-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

Modelo n-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1})$$

N-grams

Modelo bi-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

Modelo n-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1})$$

Estimación MLE: obtenemos las frecuencias desde un corpus y normalizamos.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

N-grams

Modelo bi-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

Modelo n-grama:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1})$$

Estimación MLE: obtenemos las frecuencias desde un corpus y normalizamos.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

Por razones prácticas algunas veces se usa *lop p*, ya que la suma en espacio log equivale al producto en espacio lineal:

$$p_1 \times p_2 \times p_3 \times p_4 = \exp(\log p_1 + \log p_2 + \log p_3 + \log p_4)$$

N-grams

Modelo bi-grama:

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

N-grams

Modelo bi-grama:

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

$$P(\texttt{I}\,|\,\texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam}\,|\,\texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am}\,|\,\texttt{I}) = \frac{2}{3} = .67$$

$$P(\texttt{</s>}\,|\,\texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam}\,|\,\texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do}\,|\,\texttt{I}) = \frac{1}{3} = .33$$

Evaluación de modelos de lenguaje

<u>Perplexity</u>: se basa en la existencia de una partición training/testing del corpus.

Evaluación de modelos de lenguaje

Perplexity: se basa en la existencia de una partición training/testing del corpus.

En el testing set, la perplexity es la probabilidad inversa del test set, normalizada por el número de palabras.

palabras del testing set

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}
\end{aligned}
$$

Evaluación de modelos de lenguaje

<u>Perplexity</u>: se basa en la existencia de una partición training/testing del corpus.

En el testing set, la perplexity es la probabilidad inversa del test set, normalizada por el número de palabras.

palabras del testing set

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}
\end{aligned}
$$

Usando la regla de la cadena obtenemos:

$$
\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}
$$

Evaluación de modelos de lenguaje

Perplexity en bi-grama:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Mientras mayor es la probabilidad condicional, menor es la perplejidad.

Evaluación de modelos de lenguaje

Perplexity en bi-grama:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Mientras mayor es la probabilidad condicional, menor es la perplejidad.

Si no existen dependencias, la perplejidad iguala el número de símbolos (máximo valor).

Ej.: 10 dígitos (lenguaje numérico equi-probable):

$$\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= (\frac{1}{10}^{N})^{-\frac{1}{N}} \\
&= \frac{1}{10}^{-1} \\
&= 10
\end{aligned}$$

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

Test set: 0 0 0 0 0 3 0 0 0 0

¿Cómo debiera ser la perplejidad?

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

Test set: 0 0 0 0 0 3 0 0 0 0

¿Cómo debiera ser la perplejidad? Baja, ya que el 0 ocurre muchas veces (sólo hay un símbolo infrecuente).

Evaluación de modelos de lenguaje

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

Test set: 0 0 0 0 0 3 0 0 0 0

¿Cómo debiera ser la perplejidad? Baja, ya que el 0 ocurre muchas veces (sólo hay un símbolo infrecuente).

Ej.: Se entrenan modelos unigrama, bi-grama y n-gram en el corpus Wall Street Journal (38M de tokens), con un vocabulario de 19979 términos.

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

Test set: 0 0 0 0 0 3 0 0 0 0

¿Cómo debiera ser la perplejidad? Baja, ya que el 0 ocurre muchas veces (sólo hay un símbolo infrecuente).

Ej.: Se entrenan modelos unigrama, bi-grama y n-gram en el corpus Wall Street Journal (38M de tokens), con un vocabulario de 19979 términos.

Se calcula la perplejidad en un testing set con 1.5 millones de tokens:

	Unigram	**Bigram**	**Trigram**
Perplexity	962	170	109

¿Cuál es el modelo más informativo?

Evaluación de modelos de lenguaje

Ej.: Supongamos que 0 es frecuente, y ocurre 91 veces en el training set, y los restantes 9 símbolos sólo ocurren una vez.

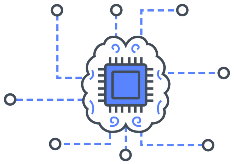
Test set: 0 0 0 0 0 3 0 0 0 0

¿Cómo debiera ser la perplejidad? Baja, ya que el 0 ocurre muchas veces (sólo hay un símbolo infrecuente).

Ej.: Se entrenan modelos unigrama, bi-grama y n-gram en el corpus Wall Street Journal (38M de tokens), con un vocabulario de 19979 términos.

Se calcula la perplejidad en un testing set con 1.5 millones de tokens:

	Unigram	Bigram	Trigram
Perplexity	962	170	109



¿Cuál es el modelo más informativo?

Suavizado de modelos de lenguaje

Muchos pares pueden no observarse en el *training set*, por lo que un modelo de lenguaje podría perder capacidad de generalización.

Suavizado de modelos de lenguaje

Muchos pares pueden no observarse en el *training set*, por lo que un modelo de lenguaje podría perder capacidad de generalización.

Otro problema relacionado a las limitaciones del training set tiene que ver con las palabras OOV (out-of-vocabulary).

Suavizado de modelos de lenguaje

Muchos pares pueden no observarse en el *training set*, por lo que un modelo de lenguaje podría perder capacidad de generalización.

Otro problema relacionado a las limitaciones del training set tiene que ver con las palabras OOV (out-of-vocabulary).

Suavizado de Laplace : agregamos una ocurrencia a todos los términos:

$$P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$$

Como se suma 1 a cada token, el denominador suma V

Suavizado de modelos de lenguaje

Muchos pares pueden no observarse en el *training set*, por lo que un modelo de lenguaje podría perder capacidad de generalización.

Otro problema relacionado a las limitaciones del training set tiene que ver con las palabras OOV (out-of-vocabulary).

Suavizado de Laplace : agregamos una ocurrencia a todos los términos:

$$P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$$

Como se suma 1 a cada token, el denominador suma V

Podemos medir el efecto en el numerador definiendo una cuenta ajustada:

$$c_i^* = (c_i + 1)\frac{N}{N+V}$$

y luego medimos el descuento relativo: $d_c = \frac{c^*}{c}$

Suavizado de modelos de lenguaje

Ej.: 9332 sentencias, V = 1446.

Testing set: ‘I want to eat chinese food lunch spend’

Suavizado de Laplace:

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Suavizado de modelos de lenguaje

Ej.: 9332 sentencias, V = 1446.

Testing set: ‘I want to eat chinese food lunch spend’

Suavizado de Laplace:

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

La relación entre perplejidad y entropía

La entropía es una medida de información definida como:

$$H(X) = -\sum_{x \in \chi} p(x) \log_2 p(x)$$

Interpretación: Cota inferior del # de bits necesarios para codificar X.

La relación entre perplejidad y entropía

La entropía es una medida de información definida como:

$$H(X) = -\sum_{x \in \chi} p(x) \log_2 p(x)$$

Interpretación: Cota inferior del # de bits necesarios para codificar X.

La entropía de todas las secuencias de palabras de largo *n* en un lenguaje *L* se define como:

$$H(w_1, w_2, \ldots, w_n) = -\sum_{W_1^n \in L} p(W_1^n) \log p(W_1^n)$$

La relación entre perplejidad y entropía

La entropía es una medida de información definida como:

$$H(X) = -\sum_{x \in \chi} p(x) \log_2 p(x)$$

Interpretación: Cota inferior del # de bits necesarios para codificar X.

La entropía de todas las secuencias de palabras de largo *n* en un lenguaje *L* se define como:

$$H(w_1, w_2, \ldots, w_n) = -\sum_{W_1^n \in L} p(W_1^n) \log p(W_1^n)$$

Definimos el *entropy rate* a nivel de palabra:

$$\frac{1}{n} H(W_1^n) = -\frac{1}{n} \sum_{W_1^n \in L} p(W_1^n) \log p(W_1^n)$$

La relación entre perplejidad y entropía

Si queremos medir la entropía de un lenguaje, debemos considerar sentencias de largo variable no acotado:

$$\begin{aligned} H(L) &= \lim_{n\to\infty} \frac{1}{n} H(w_1, w_2, \ldots, w_n) \\ &= -\lim_{n\to\infty} \frac{1}{n} \sum_{W \in L} p(w_1, \ldots, w_n) \log p(w_1, \ldots, w_n) \end{aligned}$$

La relación entre perplejidad y entropía

Si queremos medir la entropía de un lenguaje, debemos considerar sentencias de largo variable no acotado:

$$\begin{aligned} H(L) &= \lim_{n\to\infty} \frac{1}{n} H(w_1, w_2, \ldots, w_n) \\ &= -\lim_{n\to\infty} \frac{1}{n} \sum_{W\in L} p(w_1, \ldots, w_n) \log p(w_1, \ldots, w_n) \end{aligned}$$

El teorema de Shannon-McMillan-Breiman establece que si el lenguaje es regular (estacionario y ergódico), la entropía es equivalente a:

$$H(L) = \lim_{n\to\infty} -\frac{1}{n} \log p(w_1 w_2 \ldots w_n)$$

… es decir, que podemos tomar una secuencia suficientemente larga en lugar de considerar la suma sobre todas las posibles secuencias.

La relación entre perplejidad y entropía

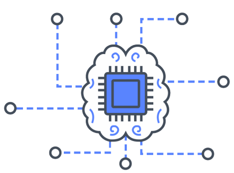
Si queremos medir la entropía de un lenguaje, debemos considerar sentencias de largo variable no acotado:

$$
\begin{aligned}
H(L) &= \lim_{n\to\infty} \frac{1}{n} H(w_1, w_2, \ldots, w_n) \\
&= -\lim_{n\to\infty} \frac{1}{n} \sum_{W \in L} p(w_1, \ldots, w_n) \log p(w_1, \ldots, w_n)
\end{aligned}
$$

El teorema de Shannon-McMillan-Breiman establece que si el lenguaje es regular (estacionario y ergódico), la entropía es equivalente a:

$$
H(L) = \lim_{n\to\infty} -\frac{1}{n} \log p(w_1 w_2 \ldots w_n)
$$

… es decir, que podemos tomar una secuencia suficientemente larga en lugar de considerar la suma sobre todas las posibles secuencias.



Estacionario: las probabilidades son invariantes en el tiempo.
Ergódico: los valores esperados de las mediciones en el sistema son iguales a las medias de muchas observaciones.

La relación entre perplejidad y entropía

Cross-entropy: se usa cuando no conocemos la distribución real *p* de los datos observados. En su lugar, usamos un modelo *m*, el cual aproxima a *p.*

$$H(p,m) = \lim_{n \to \infty} -\frac{1}{n} \sum_{W \in L} p(w_1,\ldots,w_n) \log m(w_1,\ldots,w_n)$$

La relación entre perplejidad y entropía

Cross-entropy: se usa cuando no conocemos la distribución real *p* de los datos observados. En su lugar, usamos un modelo *m*, el cual aproxima a *p.*

$$H(p,m) = \lim_{n \to \infty} -\frac{1}{n} \sum_{W \in L} p(w_1, \ldots, w_n) \log m(w_1, \ldots, w_n)$$

Para un proceso estacionario y ergódico, podemos simplificar la *cross entropy* usando el teorema de Shannon-McMillan-Breiman:

$$H(p,m) = \lim_{n \to \infty} -\frac{1}{n} \log m(w_1 w_2 \ldots w_n)$$

La relación entre perplejidad y entropía

Cross-entropy: se usa cuando no conocemos la distribución real *p* de los datos observados. En su lugar, usamos un modelo *m*, el cual aproxima a *p.*

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n} \sum_{W\in L} p(w_1,\ldots,w_n) \log m(w_1,\ldots,w_n)$$

Para un proceso estacionario y ergódico, podemos simplificar la *cross entropy* usando el teorema de Shannon-McMillan-Breiman:

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n} \log m(w_1 w_2 \ldots w_n)$$

La cross-entropy es una cota superior de la entropía. Se cumple que para cualquier modelo:

$$H(p) \leq H(p,m)$$

La relación entre perplejidad y entropía

La cross-entropy se puede definir en el límite como el largo de la secuencia infinita observada de palabras:

La aproximamos por una sentencia suficientemente larga, de largo fijo. Se aproxima por un modelo sobre una secuencia W:

$$H(W) = -\frac{1}{N}\log P(w_1 w_2 \ldots w_N)$$

La relación entre perplejidad y entropía

La cross-entropy se puede definir en el límite como el largo de la secuencia infinita observada de palabras:

La aproximamos por una sentencia suficientemente larga, de largo fijo. Se aproxima por un modelo sobre una secuencia W:

$$\longrightarrow M = P(w_i | w_{i-N+1} \ldots w_{i-1})$$

$$H(W) = -\frac{1}{N} \log P(w_1 w_2 \ldots w_N)$$

La relación entre perplejidad y entropía

La cross-entropy se puede definir en el límite como el largo de la secuencia infinita observada de palabras:

La aproximamos por una sentencia suficientemente larga, de largo fijo. Se aproxima por un modelo sobre una secuencia W:

$$M = P(w_i|w_{i-N+1} \ldots w_{i-1})$$

$$H(W) = -\frac{1}{N}\log P(w_1 w_2 \ldots w_N)$$

La perplejidad de un modelo es la *exp* de su cross-entropy:

$$\begin{aligned}
\text{Perplexity}(W) &= 2^{H(W)} \\
&= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}} \\
&= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}
\end{aligned}$$