Tarea 6: Embeddings and CNNs

Y. Sarahi Grcía González

CIMAT

Word2Vec

- Explique la estrategia de selección de paralabras dentro de la ventana de contexto en w2v, ¿Por qué se hace así y cuál es la intuición?
  El paper presentanta dos Arquitecturas 1 de red neuronal. Éstas consisten de una capa de entrda, una de proyección y una de salida. La estrategia de selección para ventana de contexto es la misma en ambos casos pero la probabilidad que se busca encontrar depende de cuál de los dos modelos se esté implementando.

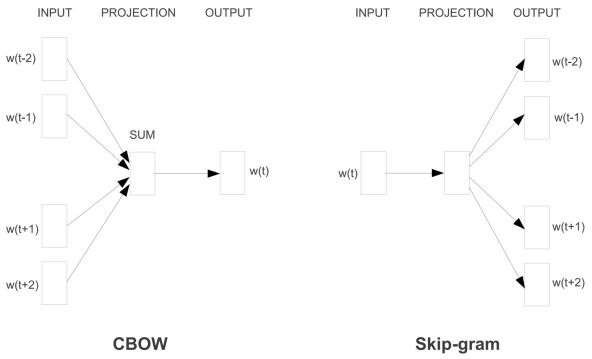


Figura 1: Arquitecturas propuestas en el paper: CBOW y Skip-gram

Se recorre cada posición en el texto, llamémosle t a la posición actual. Para esta posición t, la palabra $w(t)$ es la palabra central y, el contexto son dos palabras a la izquierda (posición $t-2$) y dos palabras a la derecha (posición $t+2$) como se muestra 2.

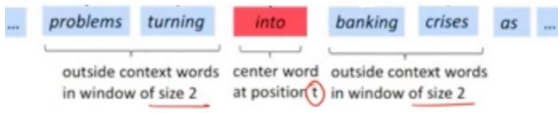


Figura 2: Ventana de contexto de tamaño dos. Palabra central en rojo y palabras *outside context* en azul.

En el caso de que la posición t sea la primera o ultima del documento, se agregan los tokens especiales de inicio/final de oración.

Cómo se puede ver en el diagrama 1, la ventana de contexto se utiliza de distinta forma en cada una de las aruitecturas. En el caso de CBOW, a partir de las palbras *outside* se calcula la probabilidad para palabra central dado el contexto que la rodea $P(w_t|w_{t-i})$ con $i \in \{-2, -1, 1, 2\}$. En el caso de Skipgram, se calcula la probabilidad de las palabras del contexto, dada la palabra central de éste contexto.

La intuición de esto sigue siendo la senántica distribucional: las palabras que se usan y aparecen en los mismos contextos tienden a transmitir significados parecidos.

- ¿Qué estrategia se usa para construir frases de palabras y construir un solo vector para conceptos basados en más de un token?
  Para aprender la representación de un sólo vector para una frase se siguen varios pasos, primero:

  1. Se encuentran las palabras que frecuentemente aparecen juntas como una frase.
  2. Se define un score 1 y un cierto umbral, de modo que todos los bigramas por encima de ese umbral se considerarán un único token.

$$score(w_i, w_j) = \frac{count(w_i, w_j) - \delta}{count(w_i) * count(w_j)} \tag{1}$$

  3. Estas frases se reemplazan por un único token (en lugar los n que conformn la frase) en el dataset de entrenamiento.

  Notemos que sólo las paralabras que aparecen juntas en uno (o unos pocos) cotexto específico serán tomadas en cuenta como un token, de modo que *Estado de México* sería uno de éstos ejemplos, pero *esto es* no lo sería. Pues las pabras *esto* y *es* podrían aparecer juntas en cualquier texto del español, mientras que, a menos que estemos hablando del estado específico, *estado*, *de* y *México* no co-ocurren.

- Según el autor de w2v ¿cuáles podrían ser las ventajas/desventajas de CBOW y Skipgram? En el primer paper (Efficient Estimation of Word Representations in vector space) se mencionan éstas dos arquitecturas y algunas otras en la sección dos *Model Architectures* y en los resultados.

  De la sección dos se puede resumir que la ventaja de los modelos propuestos por el paper tienen que ver con la complejidad. La complejidad la definen como el número de parámetros a los que se necista acceder para enterenar completamente el modelo. Y precisamente uno de los objetivos del paper es maximizar el accuracy y al mismo tiempo disminuir la complejidad computacional en sus modelos.

  En los resultados se menciona que la evaluación se hizo a través de una serie de tareas específicas, tales cómo el preguntare ¿Cuál es la palabra que es similar a "small.^{en} el mismo sentido que "biggest.^{es} similar a "big¿. Se muestran cinco tipos de preguntas semánticas y nueve de preguntas sintácticas, aunque se menciona, en total fueron 8869 semánticas y 10675 sintácticas. Para considerar una respuesta como correcta, el resultado de las operaciones algebráicas sobre los vectores debe de ser el **más cercano** a la palabra correcta.

Model Architecture	Semantic-Syntactic Word Relationship test set	
	Semantic Accuracy [%]	Syntactic Accuracy [%]
RNNLM	9	36
NNLM	23	53
CBOW	24	64
Skip-gram	55	59

Figura 3: Tabla comparativa de arquitecturas usando modelos entrenados sobre los mismos datos. Vectores de dimensión 640.

En la tabla 3 del paper w2va-1 se observa que los modelos propuesto en éste superan a las aruitecturas RNNLM y NNLM en ambas tipos de preguntas. Entre CBOW y Skipgram, el primero es mejor en la cuestión sintáctica y el segundo el la semántica (dos veces mejor). Sin mecionar, como ya se dijo antes que estas aruitecturas tiene una complejidad mucho menor.

Se mencionan dos comparaciones más bajo distintas codiciones y los resultados son similares. CBOW es mejor que NNLM y que Skip-gram en la parte sintáctica. Skip-gram es mejor que NNLM en esta tarea y es mucho mejor que todos los modelos en la parte semántica.

Las ventajas entonces se resumen en ser computacionalmente menos complejas (con la complejidad definida en el paper), además de mostrar mejor accuracy (medida con las tareas propuestas en el paper). Particularmente Skip-gram es muy superior (a CBOW y los demás modelos) en la tarea semántica.

Una desventaja que se menciona es que para poder entrenar grandes cantidades de datos, e quita la capa oculta del modelo lo que tiene omo consecuencia un modelo menos preciso, sin embargo esto se compensa precisamente con el ser más eficiente con grandes cantidades de texto. De modo que estos modelos NO son convenientes si no se cuenta con mucho texto.

- ¿Cuáles son las diferencias entre usar Hierarchical Softmax, Negative Sampling y NCE? ¿Cuál recomienda el autor y por qué?

  En el modelo de Skipgram, dada la secuencia de palabras w_1, w_2, w_T, el objetivo de éste es maximizar el promedio de la log-verosimilitud dado por

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c} log(p(w_{t+j}|w_t)) \tag{2}$$

  Con c el tamaño de la ventana de contexto. Aquí lo interesante es encontrar la probabilidad $p(w_{t+j}|w_t)$. Una manera de calcularla es con la función **softmax** 3, pero es impráctico debido a que de ésta, el costo computaciónal es proporcional al tamaño del vocabulario.

$$p(w_O|w_I) = \frac{exp(w_o * w_i)}{\sum_{w\epsilon W} exp(w * w_i)} \tag{3}$$

  Para resolver este inconveniente se usan distintas aproximaciones para calcular la probabilidad de interés, entre ellas Hierarchical Softmax (HS), Negative Sampling (NS) y Noise Contrastive Estimation (NCE). A continuación discutiremos cada una de ellas.

1. Hierarchical Softmax

   La principal ventaja sobre Softmax es que en lugar de evaluar sobre todo el vocabulario W, sólo se evaluan $log_2(W)$ nodos en la red neuronal. Representa la capa de salida como un arbol binario con W palabras como sus hijas y cada nodo es la probabilidad relativa de sus nodos hijo. Así, se puede llegar a cada palabra w por un camino desde la raíz del árbol. Sea $n(w, j)$ el j-ésimo nodo en el camino desde la raíz hasta w, y sea $L(w)$ la longitud de este camino, entonces $n(w, 1) = raiz$ y $n(w, L(w)) = w$. Además, para cualquier nodo interno n, sea $ch(n)$ un hijo fijo arbitrario de n, entonces, la softmax jerárquica define $p(w_O|w_I)$ de la siguiente manera:

$$p(w|wI) = \prod_{j=1}^{L(w)-1} \sigma\left([[n(w, j+1) = ch(n(w, j))]] \cdot v'_{n(w,j)} v_{wI}\right) \tag{4}$$

2. Noise Contrastive Estimation

   La idea básica es convertir el problema de estimar una probabilidad de palabra en un problema de clasificación binaria, lo que lo hace más manejable y eficiente computacionalmente. En lugar de tratar de calcular la probabilidad de que una palabra sea la correcta, se utiliza el enfoque de çontraste de ruido"donde el modelo se entrena para distinguir entre datos reales y datos generados artificialmente.

   NCE maximiza la probabilidad logarítmica de la softmax 3 y busca poder diferenciar entre datos y *ruido* mediante regresión logística. Sin embaergo, Skip-gram sólo se preocupa por aprender representaciones vectoriales de alta calidad, de manera que podemos simplificar siempre y cuando las representaciones vectoriales conserven su calidad.

   El ruido se genera

3. Negative Sampling Negative Sampling define como aproximación a la log-probabilidad de enteres a la ecuación 5. La tarea consiste en distinguir la palabra objetivo de las selecciones de la distribución de ruido

$$\log\sigma(v'^{T}_{w_O} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log\sigma(-v'^{T}_{w_i} v_{w_I})\right] \tag{5}$$

   La idea es eligir un subconjunto pequeño y aleatorio de palabras no vecinas (negativas) de la palabra objetivo w_o. Estas palabras negativas se muestrean de acuerdo con alguna distribución de probabilidad, como la distribución unigram, donde las palabras más frecuentes tienen más probabilidades de ser seleccionadas como ejemplos negativos.

El autor recomienda Negative Sampling (la aproximación que propone el paper) para palabras muy comunes. Además se encontraron buenos resultados usando el subsampling de palabras más frecuentes pues hace los entrenamientos más rápidos y hay una mejor representación de las palabras raras.

- ¿Cuál diría usted que es la principal conclusión y aportación del paper de w2v? ¿Qué crítica hría usted a estos papers de w2v?

  Del primer paper, para mí la principal conclusión es que se pueden obtener vectores de representación de palabras de muy buena calidad empleando un modelo muy sencillo (siempre y cuando se tenga mucho texto).

Y del segundo paper, creo que fue mostrar que las representaciones vectoriales de las palabras muestran una estructura linealque permite hacer razonamientos de tipo analogía. En este paper también se mencionó que pudieron entrenar el modelo en una cantidad de texto con muchos más ordenes de margnitud debido a lo simple de la arquitectura lo que va de la mano con la conclusión anterior.

Glove

- ¿Qué desventaja trata de solucionar de W2V?

  Lo primero que se menciona es que word2vec no utiiza adecuadamente la parte estadística del del corpus con que entrena pues se entrena en contextos locales separados que escanean todo el texto en lugar de tomar en cuenta la co-ocurrencia global de las palabras. Además a lo largo del texto menciona un par de veces que, si bien las estadísticas obtenidas de la co-ocurrencia de palabras, en w2v no es claro este hecho.

- Describa con sus propias palabras cuál es la principal estrategia para lograrlo.

  Para empezar, el nombre nos da una idea, GloVe representa Global Vectors, pues se busca capturar las estadísticas globales del texto con este modelo (a diferencia de w2v).

  Una de las ideas básicas para lograr esto es no utilizar directamente la probabilidad de co-ocurrencia $P_{i,j}$ (probabilidad de que el término j co-ocurra con el término i), sino la relación de la probabilidad de co-ocurrencia con otros términos k:

  $$\frac{P_{i,k}}{P_{j,k}}$$

  Usando esta relación, si el término k está muy relacionado con j pero no con i, el resultado será grande, y viceversa, si está muy relacionado con i pero no con j, el resultado será pequeño y cuando k esté muy relacionado o no de ambas pabras, será cercano a 1.

  Así, esta relación entre probabilidades nos da una nueva caracterización de las co-ocurrencias entre dos palabras que considera la parte local y global.

  Y de lo anterior se desprende el método propuesto, pues se plantea un modelo que, junto con las propiedades que se deben de cumplir por la naturaleza del problema y tomando en cuenta que las ocurrencias frecuencias sólo añaden ruido al problema se construye la función de costo:

  $$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right)^2 \tag{6}$$

  Donde V es el tamaño de vocabulario, X_{ij} es el numero de veces que co-ocurren los términos i y j, y f una función que cumple ser cero en cero, ser no decresiente y hacerse pequeña cuando x es grande. De este modo, da menor importancia a las co-ocurrencias más frecuentes.

  Llega a esta misma ecuación desde el enfoque de w2v también donde hace explícito el otro fallo que tiene dicho modelo (el no ser transparente con el hecho de que se utilizan estadísiticas de co-ocurrencia).

- Explique las principales conclusiones de los experimentos. Comente si cree que se logró el objetivos.

Se comentan tres distintos experimentos realizados para evaluar el modelo: Word Analogies, Word Similarity, y Named entity recognition.

El primero, según el paper, es el de mayor interés debido a que se busca encontrar una estructura vectorial adecuada. Como puede observarse en la siguiente tabla, GloVe mejora en casi todos los casos a los modelos propuestos en w2v aunque no por demasiado comparándolo con Skip-gram (resaltado en color naranja).

Se encierra en rosa el Size de los conjuntos del último bloque porue es donde GloVe señala la mayor diferencia en accuracy, sin embargo el tamaño de dataset es 7 veces más grande que SG.

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**

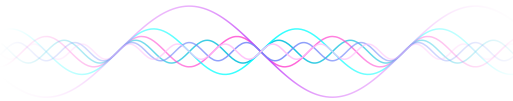
Figura 4: Tabla comparativa de modelos en tareas de tipo analogía para distintas dimensiones y tamaños de datasets.

En el segundo experimento, como se muestra en 5, GloVe también supera a los demás modelos y del tercer experimento no se menciona nada concreto.

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Figura 5: Tabla comparativa de modelos en tareas de tipo similitud de palagras para distintas dimensiones y tamaños de datasets.

Se menciona también que rendimiento es mejor en la parte sintáctica para ventanas de contexto pequeñas y asimétricas, lo cual concuerda con la intuición de que la información sintáctica se extrae principalmente del contexto inmediato y puede depender fuertemente del



orden de las palabras. Por otro lado, la información semántica es no local, y se captura mejor con tamaños de ventana más grandes.

Para el mismo corpus, vocabulario, tamaño de ventana y tiempo de entrenamiento, GloVe supera a word2vec auqnue como el mismo paper menciona, hay muchas variables que pueden modificarse en w2v y tienen un imparcto significativo que no se tomaron en cuenta en este estudio.

- ¿Encuentra alguna relación entre Glove y las clásicas TCOR y DOR? ¿Cuáles?

  Sí, pues el modelo el modelo se entrena en matrices de co-ocurrencia word-word. Además en la función de costo se toma en cuenta una función de pesado que indica la importancia que tiene la co-ocurrencia de las palabras. A aquellas que co-ocurren con demasiada frecuencia con otras palabras se les da un menor peso, pues son palabras comunes y no ayudan verdaderamente a definir otra palabra. Esto es análogo al pesado TFiDF.

- Principal conclusión y aortación del paper de Glove. ¿Qué crítica haría usted a este paper?

  Creo que la principal conclusión del paper es que las representaciones vectoriales de palabras construidas a partir de conteo de ocurrencias y las construidas con base en predicciónin no son dramáticamente diferentes a un nivel fundamental, ya que ambos investigan las estadísticas de co-ocurrencia subyacentes del corpus, pero la eficiencia con la que los métodos basados en conteo capturan estadísticas globales puede ser ventajosa. Y eso es justamente lo que busca ser GloVe: un modelo que utiliza este beneficio de los datos de conteo mientras captura las subestructuras lineales significativas (como en los métodos de predicción).

  Algo que le criticaría es que habla demasiado sobre w2v sin dar información relevante sobre cuál es la estrucutra de ese modelo. Tampoco es claro en como es el entrenamiento de su propio modelo, asumo que es análogo al de w2v pero no lo menciona.

Otros Papers

- ¿Qué desventaja trata de soluciona FastText?

  Una de las imitaciones de w2v es que trataa a cada palabra como un vector distinto y no considera la estructura interna de las palabras, lo que es una limitación importante para idiomas con una morfología rica, como el español. Algunos idiomas contienen muchas formas de palabras que ocurren raramente (o incluso nunca) en el corpus de entrenamiento, lo que dificulta aprender representaciones adecuadas para las palabras.

  En el paper proponen que es posible mejorar las representaciones vectoriales para idiomas morfológicamente ricos utilizando información a nivel de caracteres. La propuesta de FastText es aprender representaciones para n-gramas de caracteres y representar las palabras como la suma de estos vectores. De esta forma, el modelo que se propone tiene en cuenta la información de subpalabras. Esto se evaluó en nueve idiomas con diferentes morfologías, demostrando los beneficios de utilizar información a nivel de caracteres para mejorar la representación de palabras.

- ¿Cuál sería la principal desventaja de FastText sobre w2v?

  FastText es computacionalmente más intensivo debido a su enfoque de modelado de subpalabras (pues considera subpalabras además de palabras completas) y esto tiene como consecuencia que el tamaño del vocabulario sea mucho mayor. Esto implica más cálculos

durante el entrenamiento y el consumo de más recursos computacionales, especialmente en corpus de texto grandes.

Además, el modelo FastText generalmente requiere más tiempo para entrenar en comparación con Word2Vec debido a la complejidad adicional de considerar subpalabras. En el paper se menciona que utilizando n-gramas de caracteres, es aproximadamente 1.5 veces más lento de entrenar que el baseline skipgram y que procesa alrededor de 105k palabras por segundo por hilo de ejecución, en comparación con las 145k palabras por segundo por hilo de ejecución de Skipgram.

Sin embargo, creo que a pesar de estas desventajas en términos de rendimiento computacional, FastText ofrece mejoras en la calidad de las representaciones de palabras como se puede apreciar en los resultados del paper, especialmente en idiomas con una morfología rica o en tareas que involucran palabras raras o fuera del vocabulario predefinido.

- ¿Qué desventajas trata de solucionar Directional w2v y cómo lo logra? Describa las conclusiones de la sección experimental

Desventajas

De acuerdo con el paper, skip-gram tiene varias limitaciones, tales como ignorar el orden y las posiciones de las palabras. Y este contexto, se presenta el modelo Directional Skip-Gram (DSG), que aborda estas limitaciones al tener en cuenta la orientación de las palabras en el contexto para aprender embeddings de mayor calidad y menor complejidad.

Para lograr esto, DSG introduce vectores direccionales que representan la orientación de las palabras respecto a la palabra objetivo. Esto permiten al modelo capturar las relaciones semánticas y sintácticas que dependen del orden y la posición de las palabras en el contexto.

Experimentos

Se evaluó rapidez de entrenamiento, word similarity y Part-of Speech Tagging, se entrenó con la wikipedia con dos billones de tokens de palabras y se comparó contra el modelo SG original, con el Structured Skip-Gram (SSG) y con una versión simplicicada de éste último (SSSG) que se propuso en el mismo paper.

Velocidad: Comparado con el modelo original, el modelo SSG muestra una caída relativamente grande al aumentar la ventana de contexto, y una caída mucho menor para el modelo DSG. Al ampliar la ventana de contexto, la brecha de velocidad entre el modelo SSG y SG se hace más grande mientras que la brecha entre DSG y SG se hace más pequeña.

Word similarity: DSG supera a todos los modelos en corpus pequeños y en corpus grandes, sólo en un dataset queda por debajo pero por muy poco como se puede puede observar en 6:

	MEN-3k	SimLex-999	WS-353
CBOW	70.96	34.32	69.25
CWin	**74.28**	36.06	72.21
SG	71.90	34.35	70.11
SSG	71.26	31.80	69.46
SSSG	72.07	33.62	70.90
DSG	73.76	**36.10**	**72.60**

Table 3: Word similarity results ($\rho \times 100$) from embeddings trained on the large corpus.

	MEN-3k	SimLex-999	WS-353
CBOW	58.23	26.67	64.40
CWin	59.68	25.19	62.82
SG	60.19	27.14	65.23
SSG	55.42	24.00	61.95
SSSG	62.70	26.55	66.10
DSG	**63.18**	**27.51**	**66.71**

Table 4: Word similarity results ($\rho \times 100$) from embeddings trained on the small corpus.

Figura 6: Tablas comparativa de modelos en word similarity para corpus grandes y pequeñs.

- ¿Qué se dice a cerca del análisis de complejidad del Dw2v?

  Se menciona que se comparó con otros modelos de skip-gram en términos de complejidad espacial y temporal: modelo Structured Skip-Gram (SSG) y el modelo original (SG).

  También se propone un modelo simplificado al SSG, que sería el simplified Structured Skip-Gram (SSSG) donde solo modela el contexto izquierdo y derecho de una palabra dada.

  Se señala que el modelo SSG presenta una complejidad notablemente mayor en términos de espacio y tiempo cuando el contexto es más grande. Por otro lado, cada palabra en el modelo DSG solo requiere una operación adicional que la del modelo skip-gram original. Por lo quee si se amplía el contexto, el modelo DSG tendría una velocidad similar.

  La simplificada (SSSG) se presenta como una aproximación del modelo DSG dentro del marco del modelo SSG. En el lado de la salida, SSSG tiene dos vectores "wordrespectivamente para el contexto izquierdo y derecho, mientras que DSG tiene un vector "wordz un vector "direction". Como resultado, el vector de dirección de DSG se puede utilizar para predecir explícitamente si el contexto está a la izquierda o a la derecha en la predicción de palabras, mientras que SSSG no lo hace.

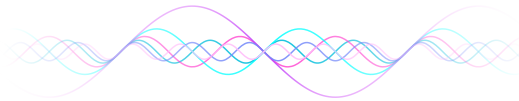
- ¿En que problemas de clasificación evaluó Kim su CNN?

  1. Clasificación de reviews positivas/negativas.
  2. Analisis de sentimiento (etiquetas:muy positivo,positivo, neutral,negativo,muy negativo)
  3. Clasificación de objetivo/subjetivo
  4. Clasificación sobre el tópico de preguntas (etiquetas:persona,lugar,informacón numérica,etc)

- En los resultados dónde estuvo involucrado algún método de clasificación con SVM, ¿Cómo fue el resultado respecto a CNNs? ¿Qué features usaba el método basado en SVM?

- En sus propias palabras, ¿Qué diferencia tienen las estrategias multi-channel y single-channel?, ¿Cuál recomienda Kim?

  Las arquitecturas de redes neuronales convolucionales (CNN) pueden tener uno o varios canales de entrada, lo que determina cómo se procesan los datos de entrada.



En la arquitectura multichannel, los canales múltiples pueden representar diferentes fuentes de información, como embbedings y/o características adicionales extraídas del texto. Cada canal puede tener su propio conjunto de filtros, y las salidas de estos canales pueden combinarse de diversas formas antes de pasar a la siguiente capa de la red.

En el caso del paper de Kim, cada canal es un set de vertores de palbabras, además se especifica que cada filtro se aplica a ambos canales y los resultados se suman para calcular las entradas del *feature map* $\mathbf{c} = [c_1, c_2 \ldots, c_{n-h+1}]$ donde n es la longitud de la oración y h el numero de palabras a las que se aplica el filtro. Aunque el filtro se aplica a ambos canales, el backpropagation se hace sólo sobre uno de los dos canales.

En el paper se recomienda usar sólo un canal pero con dimensiones extra que puedan modificarse durante el entrenamiento.

- ¿Cuál diría usted que es la principal conclusión y aportación del paper de Kim?

  Creo que la principal conclusión es que usando una arquitectura relativamente sencilla (CNN-static) se pueden obtener muy buenos resultados que compiten contra otros modelos de deep learning más sofisticados.

  Algo que no me gusto del paper fueron las conclusiones, no dicen casi nada a pesar de que se pueden concluir varias cosas a partir de la sección de resultados y discusión.