Tarea 2

'Predicción de Nacionalidad'

Word Embeddings

Alumno: Guillermo Pilleux
Profesor: Alexandre Bergel
Date: 11/10/17

Introducción

En esta tarea se hace uso directo de Word Embeddings. Un Word Embedding (WE) es la representación numérica-vectorial de las palabras que forman un lenguaje. Una de las características fundamentales y más asombrosas es que el WE es capaz de generar sentido de las palabras, calculando una medida de similitud entre ellas. De esta manera, la capacidad de un WE es inmensa, como por ejemplo, extraer el tópico general de un texto cualquiera, por mencionar una de sus utilidades.

Se utilizó un WE generado por el Profesor Jorge Pérez con 800 mil palabras fabricado con la técnica de FastText y SkipGram. Para este WE en particular, cada palabra se representa mediante un vector de 300 dimensiones, es decir, 300 valores reales que oscilan entre -1 y 1. A partir de recomendaciones del profesor y uso de esta técnica en la práctica, primordialmente por sus buenos resultados, se calcula el vector representante de una oración como el promedio de las componentes de cada palabra que la compone.

El objetivo principal de esta tarea fue predecir el origen de las personas en base a la forma en que escriben, utilizando específicamente los WE para encontrar similitudes en las nacionalidades. Para este caso particular se utilizaron dos países: Chile y Argentina.

Procedimiento

Obtención de los datos

Para cada país en estudio, se obtienen mil oraciones desde la TwiteR API por medio de R. Para el caso de Chile, se buscó directamente en Santiago y para Argentina en Buenos Aires. Los datos se guardaron en archivos de texto para realizar las consultas posteriores.

Twitter pone una limitación en cuanto a la cantidad de consultas que se obtienen, haciendo el proceso de obtención de datos un poco engorrosa.

Almacenamiento del WE en una base de dato (MySQL)

Se guarda el WE completo en una base de datos para poder consultar los vectores representantes de cada palabra de una manera directa (en vez de realizar búsquedas en un archivo de texto, que es el formato en que venía el WE). Una vez almacenado, se puede comenzar a realizar consultas sobre las palabras, pero se dio a cuenta que se demora un tiempo significante en consultar cada palabra por lo que consultar una oración completa se demoraría varios minutos.

Indexamiento del WE con Sphinx

Como el tiempo de cálculo del vector representante de una oración se convirtió en un problema, se prosigue a indexar el WE con Sphinx, un indexador bastante poderoso que disminuyó la consulta de una oración de varios minutos a unos cuantos milisegundos, pudiéndose así calcular el vector representativo de todas las oraciones obtenidas en un tiempo aceptable.

Cálculo de vector representante de una oración

Para las dos mil oraciones que se obtuvieron, se calcula el vector representante como el promedio de las componentes de cada palabra que compone dicha oración y se almacena en otro archivo de texto para luego consultar directamente el vector representativo y no hacer más cálculos intermediarios. Este procedimiento se realizó en PHP lo cual facilitó la carga comparada a haberla hecho en Java (se inició en Java, llegando a la conclusión de que Java no es el lenguaje adecuado para realizar Minería de Datos).

Palabras encontradas en el WE

La cantidad promedio de palabras encontradas en el WE utilizado por oración fue de un 80%, es decir que un 20% de las palabras de cada oración no se lograron reconocer por motivos de faltas ortográficas o simplemente que el WE no la contenía. De esta manera, se toma la decisión de no cambiar el WE debido a que la cifra promedio de palabras es considerable y representativa.

Entrenamiento y Testing

Finalmente, se crea la red neuronal y se entregan los vectores representantes de cada oración junto con un valor binario (0 o 1 que define la nacionalidad) para realizar el entrenamiento. Se utiliza el 80% del dataset para la fase de entrenamiento, realizándose 10 epocs con este conjunto de datos.

La fase de testing se utiliza el 20% restante del dataset y simplemente se verifica que el output de la red sea menor a 0.5 para validar nacionalidad Chilena y mayor a 0.5 para validar nacionalidad Argentina.

Arquitectura Neuronal

Para definir el diseño final de la red neuronal se crean distintas redes con las diferentes configuraciones que se enseñan a continuación:

Neuronas de input: 300

1. (300-2-1)
2. (300-2-2)
3. (300-5-1)
4. (300-5-2)
5. (300-2-2-1)

Para cada configuración se revisó el delta de los pesos promediados, la tasa de aserción y el tiempo de ejecución.

Las redes (1,2) y (3,4) se comportaron de forma bastante similar, por lo que se optó por la que contenía menos neuronas. Aún que entre la (1) y la (3) no existía una diferencia realmente significativa, ganó la (1) por simplicidad de la neurona de output.

La que presentó mayor diferencia en el tiempo de ejecución y el delta de los pesos, fue la (5). Esta configuración presentó un mayor delta, es decir los pesos de cada neurona se modificaron, en promedio, más que las otras configuraciones por lo que se descartó ya que poseía una mayor modificación sobre nuevos datos, lo cual no es deseado debido a que no se quiere que exista sobreajuste de los datos u olvido de la red.

El diseño final vendría siendo la red con la configuración (300-2-1), la cual se desempeñó mejor en comparación al resto en los ámbitos discutidos previamente.

Resultados

Se dividen ambos grupos de oraciones en 80-20 para entrenar y hacer testing respectivamente. Finalmente la red logra predecir con un ~95% de certeza la nacionalidad de la persona que escribió dicha oración.

Tomar en cuenta que mil oraciones por nacionalidad no es una cifra significativa, pero sí demuestra que se puede realizar esta tarea a mayor escala, refiriéndose a mayor cantidad de nacionalidades con mayor cantidad de datos (oraciones). Además, se puede tener más cuidado al momento de limpiar los datos, por ejemplo corregir errores de escritura, agregar las palabras mal escritas al WE o cambiar el WE.

Proyecciones

Quedo con las ansias de poder seguir dándole distintos usos a los WE. Primero, deseo extender este mismo proyecto a más nacionalidades y a más datos para corroborar que existe una relación y distinción directa entre la forma que hablan las distintas culturas latinoamericanas.

Lo que no se logró probar en este proyecto fue la aritmética de las palabras debido a que no se alcanzó a implementar una función de similitud para comparar el vector representativo de una oración o texto con una palabra del WE.

Los clásicos ejemplos que denotan esta última funcionalidad descrita son:

1. Rey:Hombe :: Reina:Mujer => Rey – Hombe + Mujer = Reina
   Se lee: Rey es a Hombre como Reina es a Mujer y se representa aritméticamente como Rey – Hombre = Reina - Mujer
2. Chile:Santiago :: Alemania:Berlin => Chile – Santiago = Alemania – Berlin => Berlin = Alemania – Chile + Santiago. Esta aritmética funciona con cualquier otro país:capital.