# TP Final - Simulación

## 1 Datos

**Alumno**: Tomás Sebastián Accini
**Padrón**: 99136
**Paper**: Brent Harrison, Christopher Purdy, and Mark O. Riedl, *Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks.* School of Interactive Computing, Georgia Institute of Technology Atlanta, Georgia, USA

# 2 Resumen

La presente investigación tiene como objetivo estudiar e implementar un algoritmo de generación de texto basado en cadenas de Markov. A partir del procesamiento de diferentes textos se busca un modelo matemático que simule palabras dada una o varias palabras anteriores. El paper busca aplicar el algoritmo a la generación automática de historias, que podría servir como inspiración para guiones de libros, películas o cualquier tipo de formato que transmita una historia. Para evaluar los resultados obtenidos, los mismos fueron analizados por redes neuronales que clasificaban los textos en historias y otros, entrenadas con giros argumentales extraídos de Wikipedia. Así, si un texto generado por la cadena de Markov era aprobado por las redes neuronales, el mismo tendrá características en común con otras historias. El estudio muestra que un 85% de los textos generados fueron aprobados por las redes neuronales.

# 3 Introducción

## 3.1 Cadenas de Markov

Las cadenas de Markov, llamadas así por su creador Andrey Markov, son sistemas matemáticos utilizados para simular eventos futuros basados únicamente en el estado actual. Por ejemplo, si nuestro sistema representa los días soleados y lluviosos (únicamente dos estados posibles), si hoy es un día soleado existirá cierta probabilidad p de que el día siguiente sea soleado, y cierta probabilidad 1-p de que el día siguiente esté lluvioso, y lo mismo aplica para los días de lluvia. Por lo tanto, es posible simular sucesivos acontecimientos dado un estado inicial que modele el clima de una región.

Una característica destacable de estos modelos matemáticos es la pérdida de memoria: las probabilidades de los diferentes estados para el siguiente instante solo depende del estado actual, independientemente de los estados por los que estuvo previamente. Así, si el estado actual es soleado, las probabilidades de que el próximo día sea soleado o lluvioso son iguales independientemente de si los 100 días anteriores fueron soleados o si los 100 días anteriores fueron lluviosos.

## 3.2 Aplicación a generación de texto

La teoría de las cadena de Markov es aplicable a textos. La versión más simple es considerar que cada palabra es un estado, y al analizar un texto se buscan todas las apariciones de una determinada palabra y la palabra siguiente a cada una de las apariciones. Una vez listadas todas las palabras que le siguieron a una determinada palabra en el texto, se puede calcular la probabilidad de cada uno de los siguientes estados. Por ejemplo, si tenemos el texto “decime a mi a quien acompañó a mi casa.”, la palabra “a” aparece tres veces, y es seguida de las palabras “mi” dos veces, y “quien” una vez. Por lo tanto, dado el estado “a”, los siguientes estados posibles son “mi” con una probabilidad de ⅔, y “quien” con una probabilidad de ⅓. Una vez procesado todo el dataset de entrenamiento y generados los estados, es posible seleccionar un estado inicial al azar y simular la generación de un texto.

Es importante destacar que el resultado generado estará fuertemente influenciado por el texto original, ya que las probabilidades se basan en el mismo. Las palabras utilizadas, su frecuencia, su orden, y las estructuras que se utilizan en el texto original van a determinar las características del resultado. En otras palabras, modelar una cadena de Markov con discursos de Donald Trump y otra con discursos de Hillary Clinton generan resultados fácilmente identificables, dada las diferencias entre los discursos.

### 3.2.1 Normalización del texto

Siempre que se trabaja con información no estructurada hay un fuerte trabajo previo de manipulación de los datos para llevarlo a un formato adecuado para su utilización. Es el caso de los

textos, los cuales poseen numerosas dificultades: signos de puntuación, mayúscula y minúscula, géneros, tiempos verbales, familias de palabras, singularidad, etc. Además, por la premisa de no repetir oraciones y palabras se suelen utilizar sinónimos y reformular las oraciones de maneras diferentes, por lo que los textos suelen tener una gran cantidad de palabras diferentes. A esto se lo conoce como la dimensionalidad del problema, y ocasiona no solo problemas de perfomance sino peores resultados finales, ya que la variabilidad de estados que le pueden seguir a una palabra se reduce y se cae en repeticiones forzadas.

Si bien hay múltiples estrategias para lidiar con estos inconvenientes (algunos de los cuales serán explicados en la sección de posibles mejoras), en la implementación se utilizaron las siguientes técnicas:

- Eliminar signos de puntuación
- Convertir todas las letras a minúsculas
- Eliminar los números
- Eliminar espacios extra y saltos de línea.

### 3.2.2 Tokenización

Como se ha dicho previamente, la forma más simple de modelar un texto con cadenas de Markov es haciendo que cada estado sea una palabra diferente. Sin embargo, esto puede generar texto que al leerse no tiene demasiada coherencia ya que los temas cambian rápidamente y se pierde el hilo de la narrativa. Por esa razón, una posible mejora es hacer que los estados, o **tokens** estén compuesto por más de una palabra. Esto reduce el número de posibles estados a saltar desde el estado actual, dándole mayor coherencia al texto. Como desventaja, si el número de palabras es muy grande, la variabilidad de posibles estados siguientes se reduce drásticamente, generando textos demasiado similares al original.

Tomando el caso de que los tokens estén compuestos por dos palabras, dado el texto “yo puedo jugar fútbol y además yo puedo cantar bien”, los tokens generados serían “yo puedo”, “jugar fútbol”, “y además”, “yo puedo” y “cantar bien”. Para el caso del token “yo puedo” tengo dos posibles siguientes estados: “jugar fútbol” y “cantar bien”.

En particular para este trabajo de investigación se probaron tokens de una, dos y tres palabras. Con una palabra los textos generados no tenían demasiada coherencia, mientras que con tres palabras, y dado el tamaño reducido de los datasets utilizados (lo que limitaba los posibles siguientes estados), el texto generado resultó una copia casi textual del texto original.

### 3.2.3 Cálculo de probabilidades

Una vez que tenemos generados los tokens, y ordenados en una lista en el orden de ocurrencia en el texto, simplemente se recorrió esa lista y por cada token se agregó como posible próximo estado el de la siguiente posición. Cuando todos los posibles próximos estados están listados, se calcula la probabilidad de cada uno como la cantidad de repeticiones del estado dividido la cantidad total de próximos estados.

### 3.2.4 Simulación del siguiente token

A partir de las probabilidades calculadas en el punto anterior, es posible simular un salto al siguiente estado mediante la generación de una variable uniforme [0, 1). Es necesario armar el vector de probabilidad acumulada, que consiste en que en la posición i del vector esté la suma de las probabilidades desde la posición 0 hasta la i. Por ejemplo, si en un estado A tenemos los posibles próximos estados B, C y D, con probabilidades 0.5, 0.4 y 0.1 respectivamente, entonces el vector de probabilidades será [0.5, 0.4, 0.1], y el vector de probabilidad acumulada es [0.5, 0.9, 1]. Una vez generado el valor uniforme, recorremos el vector de probabilidad acumulada hasta encontrar una posición i donde el valor del vector de probabilidad acumulada sea mayor al valor de la uniforme. Si obtenemos el valor 0.4, entonces la posición 0 es la primera que es mayor al valor generado, por lo que saltaremos al estado B. Si obtenemos el valor 0.95, la última posición será la primera que es mayor al valor obtenido, por lo que iremos al estado D.

### 3.2.5 Simulación de textos

Ya implementado el salto de un estado a otro basado en las probabilidades calculadas, es trivial generar un texto:

1. Seleccionar el estado inicial al azar
2. Iterar n veces, siendo n la cantidad de tokens a generar (no es lo mismo que la cantidad de palabras resultantes):
    a. Agregar al final del resultado el token actual.
    b. Simular el próximo token a partir del token actual.
    c. El token actual pasa a ser el token generado a partir de la simulación.
3. Generar el texto a partir de la lista de tokens obtenida.

## 3.3 Evaluación de resultados

La validación de la generación de textos es una tarea difícil por la subjetividad de la misma. Se puede tener en cuenta diferentes aspectos del resultado: coherencia, calidad, fluidez para leer, tasa de repetición de palabras, conjugación correcta de los verbos, correcta gramática, sintaxis, parecido con el texto original (por ejemplo, para generar discursos presidenciales de un candidato), etc. Como se observa, no es una evaluación objetiva y concreta, que pueda ser fácilmente medible en valores numéricos y comparables entre sí. Por eso, se suele recurrir a dos métodos de evaluación: la validación manual y la validación mediante redes neuronales.

### 3.3.1 Validación manual

Debido a la alta complejidad para establecer un criterio claro y objetivo de validación, los textos generados se validaron y evaluaron de forma manual y subjetiva. Así, se tuvo en cuenta principalmente la coherencia de los textos originados, los temas tratados y las similitudes con el texto original.

### 3.3.2 Redes neuronales

En el paper se propone la utilización de redes neuronales entrenadas con giros argumentativos extraídos de wikipedia, para luego clasificar las historias obtenidas como historias válidas o no, logrando un 85% de índice de aprobación. Dado que este tema excede los conocimiento obtenidos, y que no se detalla la implementación y set de entrenamiento de estas redes neuronales, no se realizó la implementación de este método de validación.

# 4 Resultados

En la presente sección se explicará la implementación realizada, para luego entrar en detalle de los datasets utilizados y los resultados obtenidos.

## 4.1 Implementación propia

A partir de los textos de los datasets, se normalizaron y tokenizaron los datos, se procesaron los tokens para cargar las relaciones entre los mismos para luego calcular las probabilidades de los siguientes estados posibles. Luego, se simularon n tokens a partir de uno al azar.

En cuanto a la normalización de los textos, primero se pasaron todos los caracteres a minúscula para reducir la dimensión del problema. Luego, se removieron todos los signos de puntuación (comas, puntos, signos de exclamación, signos de pregunta) y los saltos de línea y espacios extra. Además, se eliminaron todos los números ya que, por la naturaleza del problema, no tenía sentido agregarlos a las historias generadas debido a que los valores son muy particulares de la oración a la que pertenece.

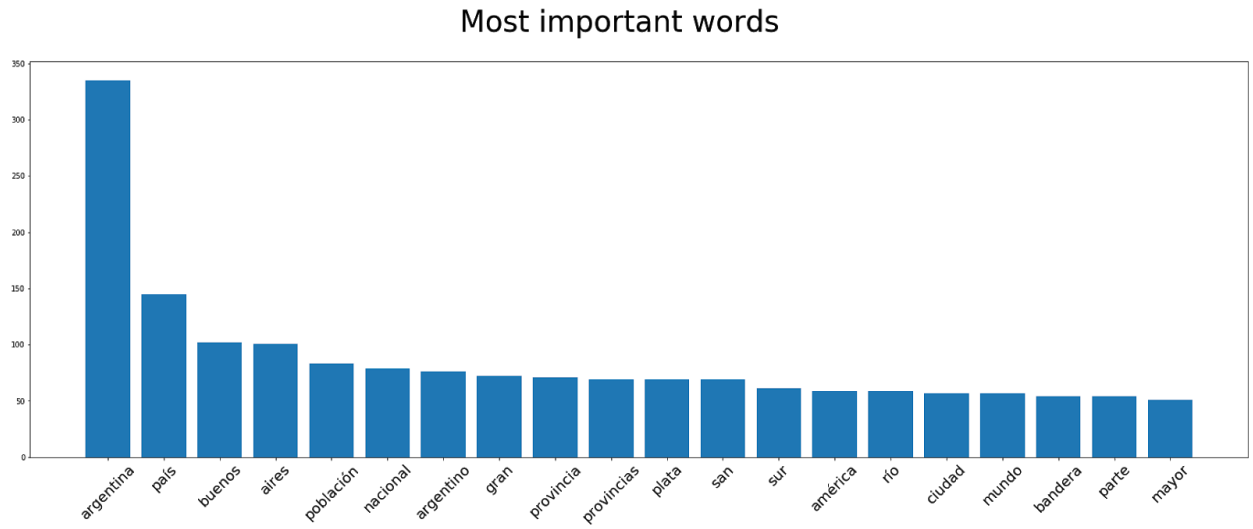
Posteriormente se hizo una primera separación en palabras, preservando el orden de las mismas. Para hacer la tokenización, la implementación permite establecer la cantidad de palabras que forman un token. Así, se pudieron con múltiples valores, obteniendo resultados muy diversos según el número elegido. Como ya se explicó en la introducción, si un token está compuesto por pocas palabras la oración generada suele tener menos coherencia, pero mayor flexibilidad para generar temas novedosos, ya que hay una mayor variabilidad en los posibles siguientes estados. Al mismo tiempo, si los tokens son generados por una cantidad muy grande de palabras, se reduce drásticamente la cantidad de posibles siguientes estados, generando textos muy similares al original. En particular para los datasets analizados, se probaron con 1, 2 y 3 palabras por tokens, con resultados similares en todos los casos: con 1 no se formaron oraciones coherentes, mientras que con 3 eran citas textuales del texto original. Con dos palabras por token se obtuvieron los resultados más interesantes.

Además, se hizo un breve análisis de cada uno de los datasets, incluyendo la cantidad de palabras (totales y únicas), y gráficos que mostraban las palabras más importantes del texto (siempre excluyendo una lista de palabras comunes según el idioma, que se consideraron que no aportan valor extra al análisis).

## 4.2 Datasets utilizados y resultados

### 4.2.1 Página de Argentina de Wikipedia

El dataset consiste en la página de Argentina en español, de Wikipedia. El mismo cuenta con 31.523 palabras, de las cuales 5.937 son diferentes entre sí. Sacando las palabras clásicas del idioma español, las que más aparecieron en el texto fueron:

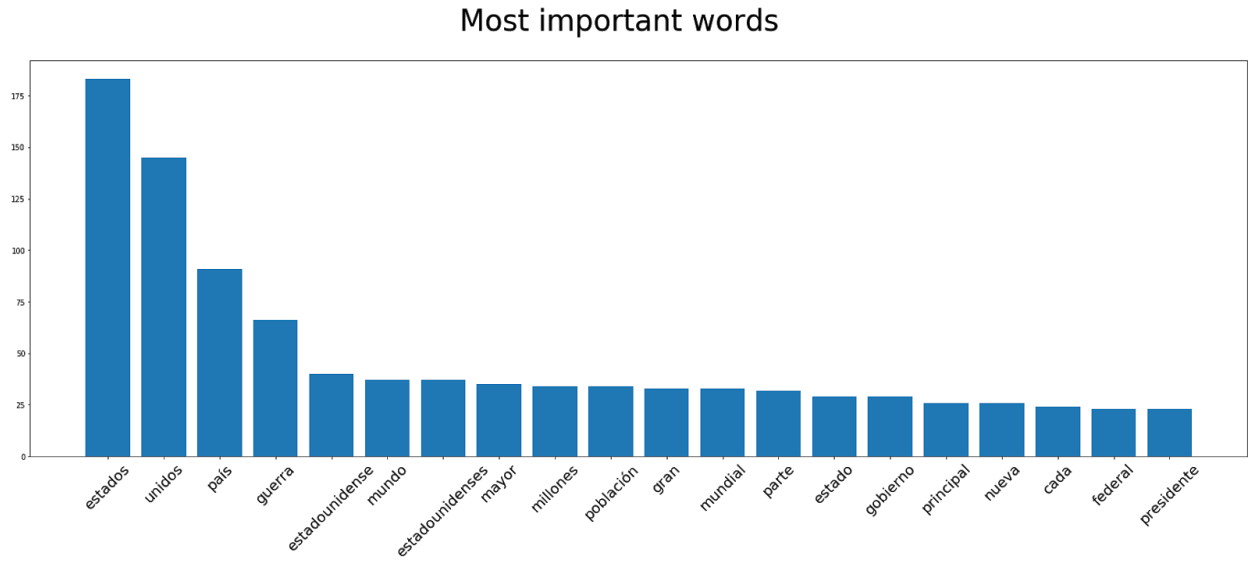



Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
  - “alvear el objeto de américa del país latinoamericano en la provincia de infraestructura entre otros”
  - “espaciadas entre del precepto alberdiano de un país se vota por vicente lópez con los”
  - “épico se encuentra en la renacionalización de gobernar es el instituto balseiro ubicado en bariloche”
- Con dos palabras por token:
  - “las letras del tango pero ha perdido buena parte de la alianza fue encabezada por el indec las provincias sus”
  - “tanto públicas como privadas que se desarrollan también construye helicópteros maquinarias agrícolas produce el ciclo completo de la soja la”
  - “ascendencia europea es menor y que constituye uno de los hombres obtuvieron el título más importante de la historia argentina”
- Con tres palabras por token:
  - “llamados comechingones resistieron con éxito la invasión incaica y se mantuvieron como señoríos independientes a”
  - “español andino se fusiona con el dialecto de rioplatense la provincia de buenos aires con”
  - “leche es muy importante consumiéndose alrededor de litros por persona por año de la existencia de grandes disponibilidades de leche se ha derivado un alto consumo de alimentos derivados como”

### 4.2.2 Página de Estados Unidos de Wikipedia

El dataset consiste en la página de Estados Unidos en español, de Wikipedia. El mismo cuenta con 14.425 palabras, de las cuales 3.428 son diferentes entre sí. Sacando las palabras clásicas del idioma español, las que más aparecieron en el texto fueron:

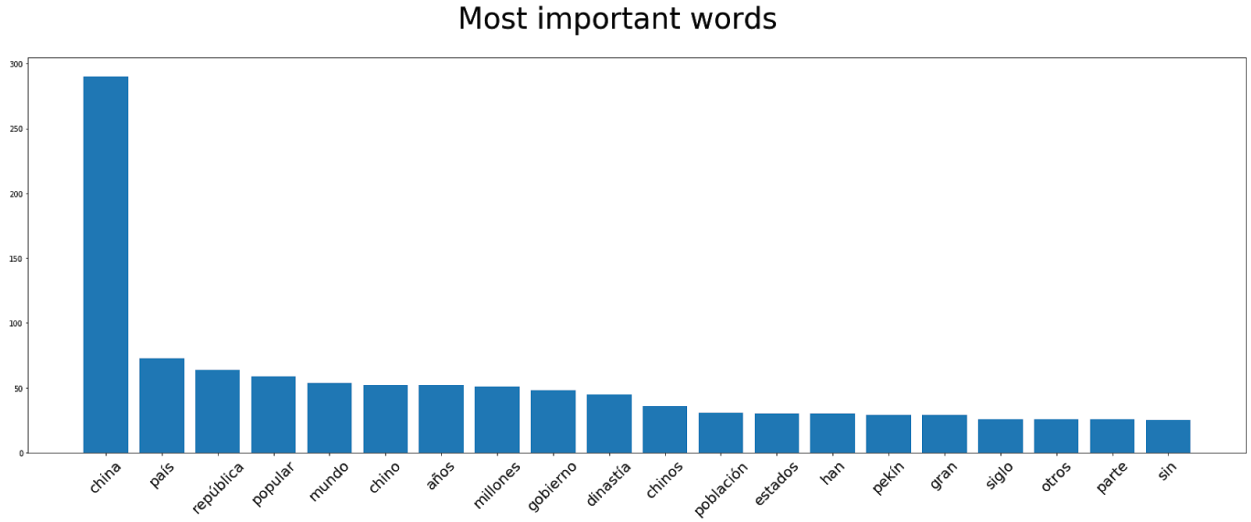



Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
  - “reubicación de materiales y públicas de igual forma la música artículos principales carreras de julio”
  - “cometieron el de influencia de los logros socioeconómicos de los aliados contra el estado islámico”
  - “locales ocupaban el estado libre asociado con el feto ya que destacan autores como estado”
- Con dos palabras por token:
  - “comercial estadounidense era de millones de afroamericanos que habían sido esclavos convirtiéndolos en ciudadanos y dándoles el derecho consuetudinario en”
  - “mundo aunque en términos de gasto per cápita de toneladas de petróleo al año en salarios aunque su legalidad está”
  - “transporte de mercancías por ferrocarril es muy importante relativamente pocas personas utilizan el transporte público para acudir al trabajo un”
- Con tres palabras por token:
  - “y al presidente de los estados unidos donde se reúne el congreso estados unidos es una nación multicultural hogar de una amplia variedad de grupos étnicos tradiciones y valores aparte”
  - “suelen emplearse de manera correcta la abreviatura ee uu estados unidos o la sigla eua estados unidos de américa aunque frecuente en español es incorrecto emplear la sigla inglesa usa”
  - “es considerado un país megadiverso unas especies de plantas vasculares viven en los estados unidos cerca del de los universitarios asisten a colleges públicos como la universidad de virginia un”

### 4.2.3 Página de China de Wikipedia

El dataset consiste en la página de China en español, de Wikipedia. El mismo cuenta con 15.515 palabras, de las cuales 3.599 son diferentes entre sí. Sacando las palabras clásicas del idioma español, las que más aparecieron en el texto fueron:

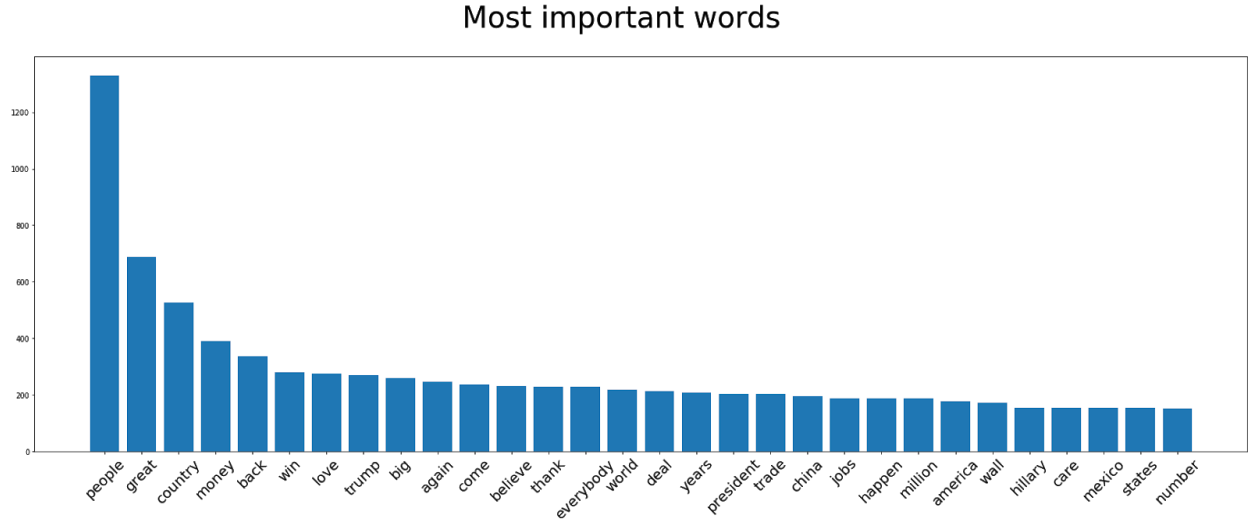



Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
  - “fin del arroz el segundo miembro de taiwán el kuomintang liderado por ejemplo de pekín”
  - “incrementó los yuan y animales y los años atrás una vez más poblado del planeta”
  - “septiembre de las dos hijos si lo que mantuvieron un el jengibre el muy criticado”
- Con dos palabras por token:
  - “área territorial después de la creación de la onu china tiene una tasa de mortalidad infantil es de por cada”
  - “transitados en el mundo con y millones de chinos en las zonas rurales no tienen acceso a agua potable y”
  - “ejército chino transporte el puente de donghai es uno de los mejores baloncestistas de china carreras de barcodragón un deporte”
- Con tres palabras por token:
  - “mundial y la retirada de sus tropas de china el partido comunista de china pcc cuyo poder está consagrado en la constitución la constitución de la república popular china están”
  - “en términos de pib medido en paridad de poder adquisitivo y manteniéndose como la segunda potencia por pib nominal china es además el mayor exportador e importador de bienes y”
  - “un país que tiene armas nucleares reconocidas china es considerada una potencia militar regional y una superpotencia militar emergente de acuerdo al informe de del departamento de defensa de estados”

### 4.2.4 Discursos de Donald Trump en las elecciones estadounidenses

El dataset consiste en discursos de Trump en las elecciones presidenciales del 2016 de los Estados Unidos. El mismo cuenta con 164.789 palabras, de las cuales 6.012 son diferentes entre sí. Sacando las palabras clásicas del idioma inglés, las que más aparecieron en el texto fueron:

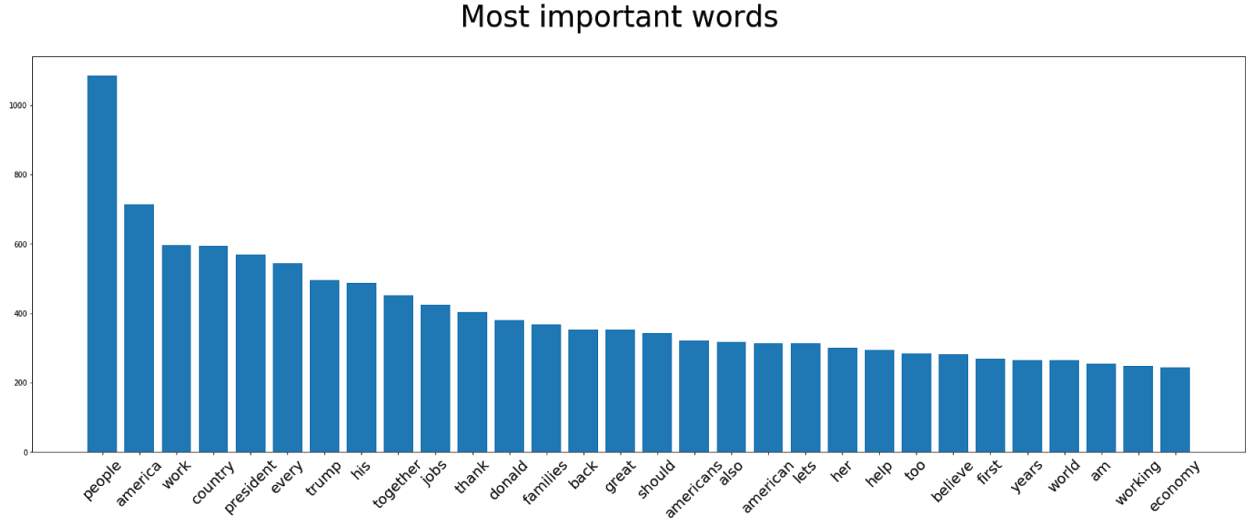



Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
  - “global warming which are they are in by the radar nobody recognizes them iraq was”
  - “upper income earned abroad in a picture because were going to bring back in the”
  - “henry right now he spends million on something because of words radical islam is standing”
- Con dos palabras por token:
  - “any deals we all saw and we witnessed something that you could say well were going after hillary clinton she”
  - “means less than zero and i said it very strongly in the meantime in michigan and other places and they”
  - “though he was under pressure because theres nothing we have to get rid of the bullets went the opposite direction”
- Con tres palabras por token:
  - “four or five weeks ill tell you but they got access so they signed a pledge they will support i dont want their support it wont mean one vote than”
  - “world take advantage of us both militarily and we dont win anymore we dont win anymore we dont win with trade we dont win we dont win at anything i”
  - “extent is the power of weaponry its the power its the tremendous power you know years ago i mean it took them like years to build it so you know”

### 4.2.5 Discursos de Hillary Clinton en las elecciones estadounidenses

El dataset consiste en discursos de Hillary Clinton en las elecciones presidenciales del 2016 de los Estados Unidos. El mismo cuenta con 209.507 palabras, de las cuales 9.602 son diferentes entre sí. Sacando las palabras clásicas del idioma inglés, las que más aparecieron en el texto fueron:

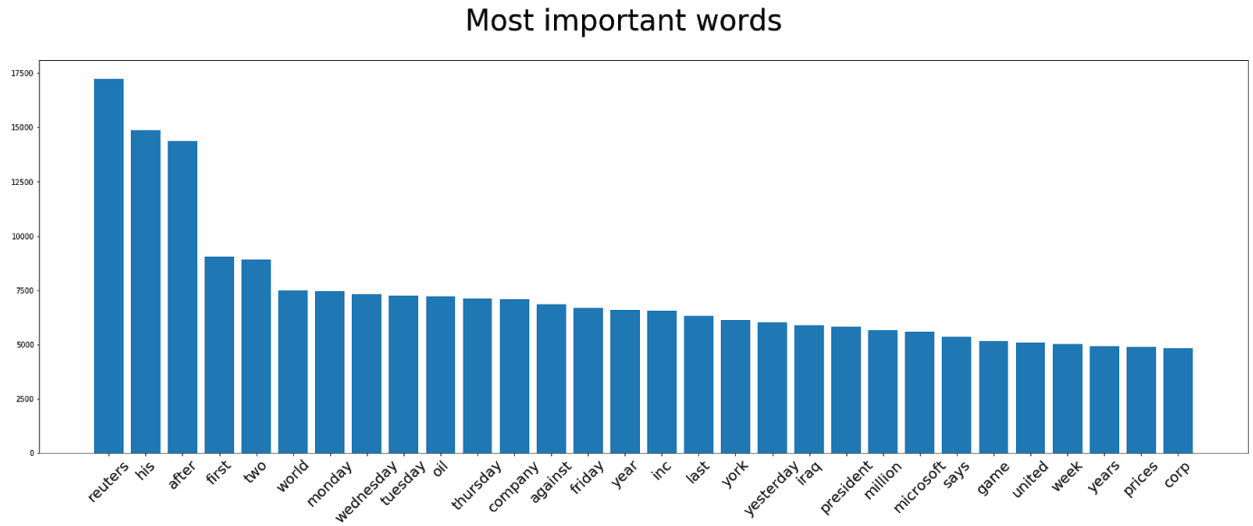



Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
  - “youre wrong with we talk about illegal immigration on that brexits going on a statement”
  - “reaping with honor of public schools and we might have more good middle class i”
  - “shortchange her own health care of job is the hatefulness attacks one has the armed”
- Con dos palabras por token:
  - “are within reach where families are supported streets are safe and yes we have created with all of this on”
  - “work drive hundreds of campuses across the country attacks that disproportionately affect lowincome voters people of color students the elderly”
  - “off donald trumps comments the textbook definition of a racist a person filled with hatred with an assault weapon we”
- Con tres palabras por token:
  - “four or five weeks ill tell you but they got access so they signed a pledge they will support i dont want their support it wont mean one vote than”
  - “world take advantage of us both militarily and we dont win anymore we dont win anymore we dont win with trade we dont win we dont win at anything i”
  - “extent is the power of weaponry its the power its the tremendous power you know years ago i mean it took them like years to build it so you know”

### 4.2.6 Recopilación de noticias

Analizamos un gran dataset de noticias de diferentes medios de comunicación estadounidenses pertenecientes al aglomerado Reuters, en inglés. El mismo cuenta con 4.406.922 palabras, de las cuales 90.838 son diferentes entre sí. Sacando las palabras clásicas del idioma inglés, las que más aparecieron en el texto fueron:




Algunas muestras de oraciones generadas a partir de este dataset, según el número de palabras por token:

- Con una palabra por token:
    - “plunk down threats as a search platforms microsoft will need to make sure some jewish”
    - “syndicate was the chinese officials said on the principle of plotting a third avenue stores”
    - “nonsectarian cards and international boxing final quot ordeal ugly and explore the whole truth about”
- Con dos palabras por token:
    - “by stepping up quota compliance to thwart a takeover investors still got to celebrate a historic development in the computing”
    - “sunday liberals celebrate as ukraine waits poll result reuters reuters us house has approved a stent inserted wednesday into an”
    - “blade a spanish supercomputer built by china and india the shift would wreck mideast peace efforts the bush administration dramatically”
- Con tres palabras por token:
    - “enough by kristen philipkoski got antinuke pills probably not potassium iodide pills could ward off cancer in the event of a nuclear accident but many states have refused to take”

- “are more exposed than ever to what could be a final step on the long road to bringing the global pact into force dogs sniff out bladder cancer dogs can”
- “league arsenal dropped five points behind chelsea in the summer the dutch side has scored goals in the final minutes to beat hamburg sv jol wants fulltime job at tottenham”

# 5 Conclusiones

La técnica de generación de texto basado en cadenas de Markov funciona, pero tiene grandes limitaciones. Desde ya, es difícil generar texto que sea entendible (y aún más, publicable) sin necesidad de modificar la salida para hacerla más clara y coherente. Sin embargo, si la idea es buscar inspiración, tal como lo plantea el paper, puede ser una buena herramienta. Si se contara con un dataset muy amplio sobre un tema de interés, es posible generar buenos cruces de ideas que pueden llegar a servir como un puntapié inicial a una idea mayor y más elaborada, al mezclar conceptos distantes en el texto pero que pueden lograr cierta coherencia. Otro de los desafíos que enfrenta esta técnica es encontrar un valor aceptable para el parámetro que indica cuántas palabras componen cada token.

## 5.1 Overfitting

En la implementación que se hizo, se puede observar que si bien se mostraron los resultados de textos generados con tokens de una, dos y tres palabras, los resultados fueron bastante contundentes. Con una palabra por token se dificulta encontrar resultados coherentes, mientras que con tres palabras por token la dimensión del problema se reduce drásticamente, obteniendo citas textuales del dataset de entrenamiento. Este último punto podría mejorarse con datasets mucho más grandes que permitan mayor variabilidad en las opciones de los posibles estados. Por lo tanto, la cantidad de palabras por token óptima está íntimamente ligada al tamaño y variabilidad del dataset.

## 5.2 Tamaño de los datasets

El tamaño de los datasets utilizados fue variado: desde datasets de 14.000 palabras a uno de 4.4 millones de palabras. Se nota una mejora en la calidad de la salida con datasets más grandes, en particular para tokens con 2 y 3 palabras (con tokens de una palabra se dificulta aún más lograr resultados coherentes).

# 6 Posibles mejoras

## 6.1 Stemming y lemmatization

Si el objetivo no es generar texto coherente, sino obtener resultados que puedan servir como inspiración para futuros trabajos, entonces las técnicas de stemming y lemmatization pueden ser de gran ayuda. El objetivo que tienen es reducir las palabras a su raíz, para poder unir múltiples palabras en una, simplificando el dataset y logrando reducir drásticamente la dimensión del problema. Por ejemplo, podríamos reducir las palabras 'corriendo', 'corrí' y 'corren' a la palabra 'correr'. Si bien hay múltiples librerías para hacer esto, la más conocida es nltk o Natural Language Toolkit, disponible para Python.

La contra que tiene utilizar técnicas del estilo es que los textos generados deben ser modificados a posteriori para generar oraciones coherentes, ya que contendrá únicamente raíces de palabras, por lo que quita claridad a los resultados inmediatos.

## 6.2 Clasificación de palabras

Una posible mejora es generar una clasificación de las palabras del dataset según su tipo: sustantivos, adjetivos, verbos, adverbios, etc. Así, sería posible obligar (o bien aumentar la probabilidad) al resultado generado a tener cierto formato. Por ejemplo, podríamos buscar generar oraciones del estilo <sujeto> <verbo> <adverbio> <objeto>. Puede ser útil cuando se busca generar múltiples oraciones cortas, pero es innegable lo laborioso que resulta el hecho de clasificar cada una de las palabras de un dataset. Podría hacerse una prueba de concepto con un dataset muy reducido, pero los resultados no serían significativos debido a la poca variabilidad de palabras.

## 6.3 Agregar símbolo que represente el final de la oración

Una de las contras que tiene la forma en que se normalizaron y tokenizaron los datasets es que genera un stream de palabras sin aparente fin. En otras palabras, no se generan oraciones concretas sino que se concatenan n palabras, con un n a gusto del que utiliza la herramienta. Si se buscara generar textos que contienen oraciones bien definidas, se podría agregar un símbolo especial, que para facilitar su explicación lo llamaremos END. Así, todo punto, signo de exclamación y signo de pregunta (y cualquier otro signo que signifique un fin de oración) será reemplazado por el símbolo END, que será parseado como una palabra más. Así, cuando una palabra es la última de una oración, uno de los posibles siguientes estados será el símbolo END, que en el resultado final puede ser mostrado como un punto. Esto agregaría un cierre a una secuencia de palabras y las englobaría en una oración.

# 7 Referencias

Los datasets fueron extraídos de las siguientes fuentes:

- Argentina: https://es.wikipedia.org/wiki/Argentina
- Estados Unidos: https://es.wikipedia.org/wiki/Estados_Unidos
- China: https://es.wikipedia.org/wiki/Rep%C3%BAblica_Popular_China
- Trump: https://github.com/ryanmcdermott/trump-speeches/blob/master/speeches.txt
- Clinton: https://votesmart.org/candidate/public-statements/55463/hillary-clinton?speechType=1#.XR_krJNKjOQ
- Noticias: https://catalog.ldc.upenn.edu/LDC97S44