

GRAS: Generating Recipes with an Algorithmic Sensibility

Machine Learning project

Introduction

- ▶ GRAS es un **algoritmo de generación de recetas**.
- ▶ Dada una lista de ingredientes de entrada (al menos uno), y si uno quiere un tipo de receta, GRAS va a proponer una receta inventada desde herramientas de un conjunto de datos de recetas.
- ▶ GRAS usa un modelo generativo **basado en probabilidades** para inventar las recetas.

Recuperación de los datos

- ▶ Quería trabajar en **francès**, porque cuando hablamos de cocina, no tiene sentido hacer de otra manera. ;-)
- ▶ No existen conjuntos de datos abiertos.
- ▶ **Web scraper** en Python para recuperar las recetas.
- ▶ Salieron 60,000 recetas y 2046 ingredientes.



```
1 {"instructions": ["...",  
  "...", "..."], "isVegetarian":  
  true, "dishType": "...",  
  "link": "...", "title": [  
    "...", "ingredients": [  
      "...", "...", "..."]}]}
```

Preprocesamiento

- Datos **escritas a la mano**, no bien formateadas. Se necesitó un preprocesamiento.
1. Poner todo en "lower case".
 2. Quitar los acentos franceses.
 3. En los ingredientes: dividir entre cantidad, unidad y ingrediente.
 - a Quitar los "stopwords" en los ingredientes.
 - b Coger la cantidad.
 - c Coger el unidad.
 - d Uniformar los nombres de ingredientes ("pomme- "pommes") con un index de stems (figura 1).

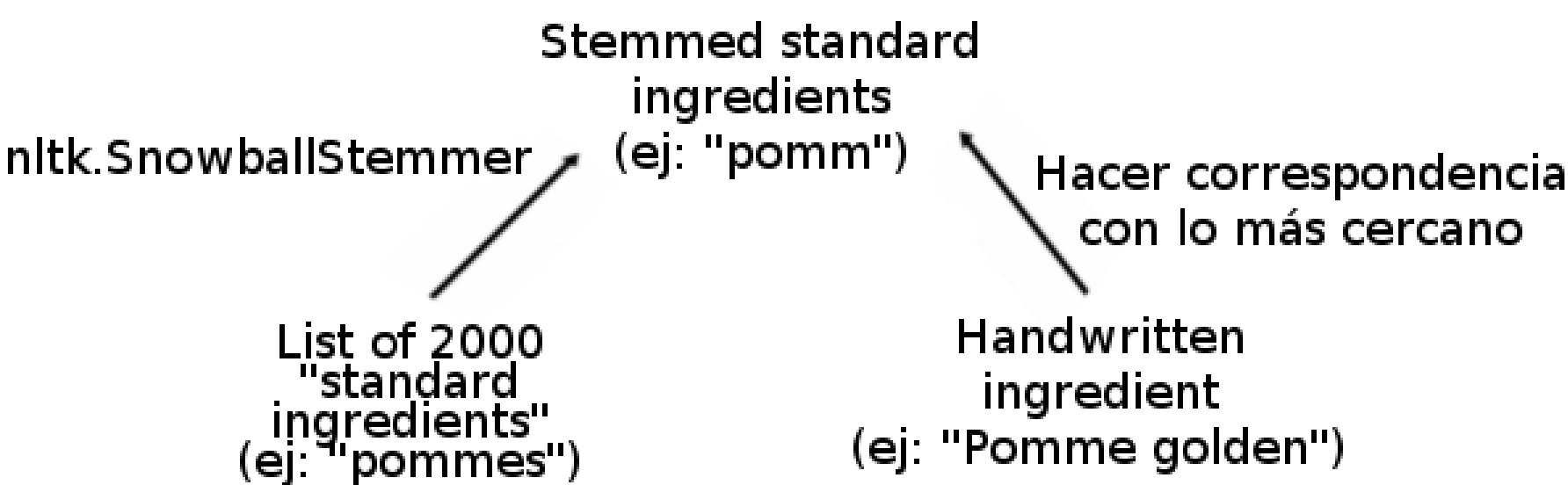


Figura: Index sobre los "stem"

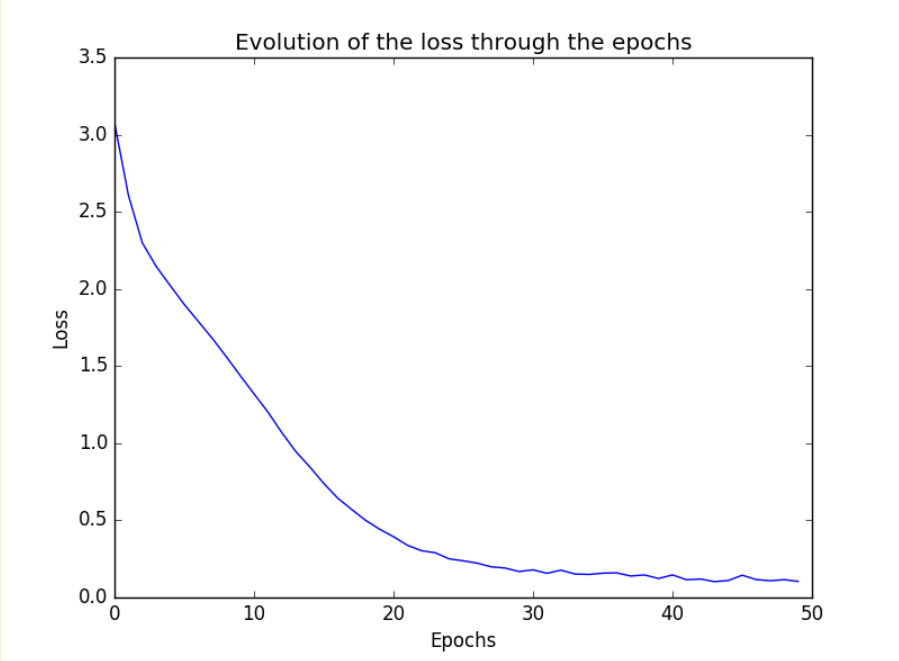
Generación de título de receta

"Tarte fagilesagne à la à la sucre et chocolat"
Generación de texto con una Red Neuronal LSTM usanda Keras.

Layer (type)	Output Shape	Param #	Connected to
lstm_1 (LSTM)	(None, 128)	90112	lstm_input_1[0][0]
dense_1 (Dense)	(None, 47)	6063	lstm_1[0][0]
activation_1 (Activation)	(None, 47)	0	dense_1[0][0]
Total params: 96175			

Train

Con todos los títulos de las recetas.



Generación

Generar empezando con los ingredientes de la receta. Cogemos la frase más representativa.

Generación de ingredientes

¿Cuáles son las entradas del algoritmo?

```
./ gras -i "<ingred1>,<ingred2>,<ingredN>"  
-c "<tipo_comida1>,<tipo_comida2>,<tipo_comidaM>"  
N ingredientes en entrada [1 mínimo]  
M tipos de comida (postre, plato principal, entrada...) [opcional]
```

¿Cuántos ingredientes vamos a generar?

Media de la cantidad de ingredientes que tienen las recetas que contienen los ingredientes de entrada.

¿Cómo encontremos ingredientes que caen bien juntos?

Modelo generativo basado en probabilidades de Bayes.
I el conjunto de ingredientes ya presentes en la receta, i_{new} el nuevo ingrediente y T los tipos de comidas dados.

$$P(i_{new}|I, T) = \frac{P(i_{new} \cap I \cap T)}{P(I \cap T)}$$

¿Cómo validar los resultados? ¿Porque el sabor no se calcule!

- ▶ Clasificador para saber si un registro **viene de los datos o es generado**.
- ▶ Calcular el **mínimo Root Mean Squared Error** entre el conjunto de ingredientes generado y cada receta en los datos. Intentamos tener algo muy cerca de 0, sin tener 0 (sino significa que la receta ya existe!). *Esta técnica esta desarrollada.*

Resultados

```
$ ./ gras -i "Boeuf, Farine" -c "PlatPrincipal"  
IN: ["Boeuf", "Farine"] ["PlatPrincipal"]  
OUT: Boeuf, Farine, Beurre, Bouillon, Tomate, Pinot  
noir, Madere, Bouquet garni, Pomme, Oignon, Carotte,  
Ail  
Title: Tere aux pommes lr sercin migrais de a la  
noix de coco faton vin gres le nois cocrons brine  
carrt boeuf, farine, beurre et bouillon.  
Validation error: 0.0221
```

- ▶ **Lista de ingredientes** coherente y un poco original (carne y frutas, vino rojo y blanco). Porque no, se puede intentar!
- ▶ **El título** de la receta no tiene ningún sentido. No tiene relación con la lista de ingredientes y hay muchos palabras inventadas.

Conclusiones y trabajos futuros

- ▶ **Los modelos generativos** forman un área de Aprendizaje de Máquina muy interesante pero muy diferente de **los modelos discriminatorios**. Se piense al revés: *no es clasificar un registro, pero crear un registro bien clasificado*.
- ▶ Hay menos literatura sobre modelos generativos, y menos técnicas conocida. Entonces son **problemas difíciles** que necesitarían mucho tiempo, porque **siempre se pueden mejorar**.
- ▶ Es un problema que **queda mucho al apreciación del desarrollador** porque no hay un objetivo claro sino "generar buenas recetas". Entonces da una dificultad más.
- ▶ En trabajos futuros, podríamos usar Adversarial Networks para generar los ingredientes.
- ▶ Y hacer una generación de instrucciones.
- ▶ Mejorar desempeño utilizando Principal Component Analysis (PCA).

Referencias

- 📖 F. Gers, "Long short-term memory in recurrent neural networks," Ph.D. dissertation, Universität Hannover, 2001.
- 📖 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.