Fake Italian Restaurants Evaluator - FIRE [GitHub]

Andrea Piolini, Srijon Mukhopadhyay, Matteo Bucalossi

1. Introduction

One of the easiest ways for Italians living abroad to check whether an Italian restaurant might not be authentic is to check if the Italian names of the dishes listed on the menu are poorly written (typos, misspellings, wrong gender agreement). This project seeks to build an NLP program that helps non-Italian speakers check whether there are many typos in a menu and assigns an "authenticity score" based on the number of typos and misspellings. If this number is high, the score will be low and the restaurant most likely is not authentic.

In particular, this project focuses on identifying whether there are any misspelled words or typos in the analyzed menus. Once we consolidated the recipes and menus in the two background and test corpora, we compared them using sets intersection to identify whether there are poorly written Italian words contained in the test corpus.

Finally, we ranked the non-Italy based Italian restaurants from 0 to 5 - 5 being authentic Italian - depending on the number of typos and misspellings identified. The "authenticity score" is 5 if the menu contained 0-5 mistakes, 4 if it had 6-10, and so on.[1]

2. Background

Research has been producing spelling check systems (particularly for English), using a mix of rule-based and statistical approaches. For instance, Manu Konchady in "Detecting Grammatical Errors in Text using a Ngram-based Ruleset" uses NTLK's corpus to check for grammar mistakes and typos. Libraries such as Enchant and SymSpell also adapt edit-distance and other algorithms for correcting errors. Yet, their main goal is to identify as well as correct mistakes (as

1 Refer to Appendix A for more details on the scoring system.

edit distance also does). In our case, we are simply trying to count mistakes to label our data. This task is indeed simpler, but other areas we encountered resulted much more unexplored:

- The literature lacks work on cross-lingual spelling check, namely a scenario where terms from different languages may be mixed in the same sentences or documents.
- Italian models and available corpora, although present mainly because of the work of the ItaliaNLP Lab of Pisa, do not include the food-specific terms we needed - hence the need to create our own custom corpus from menus and recipes.

3. Scope

First of all, we created a background corpus which is made of roughly 200 recipes and menus from GialloZafferano, an Italian recipes website, and menus retrieved from Italy-based Italian restaurants. We also created a sample test corpus containing 10 menus retrieved from Italian restaurants based in the United States. All the menus retrieved online were scraped from OpenTable, an online restaurant-reservation service company. For the background corpus, we also scraped some PDF menus using Tabula, a PDF scraper freely available online.
Once we obtained our text data in an hash table format such as json, our goal was to identify possible mistakes in the test corpus. To do so, we adapted techniques from information retrieval, such as inverted indexes, to understand peculiarities and components more prone to mistakes. We created hash tables for the frequencies of terms in each document as well for the closest neighbors of each term, and another table at the documents-level. The tabular format allows for easier applications of well-tested algorithms such as edit distance and cosine similarity to compare and analyze background vs test corpus.

4. Outcome

The scraping took a fair toll on our workflow, and we had to create custom scrapers for each different website and pdf parsing did not allow to process pdf documents on a large scale. On the other hand, our first attempts at analyzing and getting results from such data did not

succeed. The main obstacle for a deeper analysis of the text was indeed the lack of sentences in our data (as we extracted mainly dishes and ingredients), making grammatical analysis virtually impossible. Thus, different attempts to work on a vec2vec model to process the tokens could not be used in a meaningful way, and the original intent to assess test tokens by their word probability calculated from n-grams did not yield any fruitful result either, as we could not get proper n-grams from merely dishes names in menus.

Thus, basing our approach on the fundamentals of many spell checkers widely used, we proceeded to focus our attention on the corpora of words and provided the most accurate score possible by intersecting them. This approach allowed us to provide the authenticity score by simply looking at the words used by the restaurants, and, assuming a compelling background corpus and a clean test menu, such a score would be fairly accurate.

The result obtained on the sample test corpus matched closely our expectations.[2] Although some particular terms (mainly dialectal or very unique in both English and Italian) were wrongly identified as mistakes, false positives were well below a 50% threshold in our opinion and we believe our scoring system will generally approximate a faithful result.

5. Challenges

- **Data acquisition.** It proved to be rather complicated as there are no existing public corpora containing the words we needed. We could successfully scrape and clean raw data from the internet; yet, the process was time consuming and tricky at times.
- **Unwanted English words.** When we created the test corpus, we had to remove all the English words from it otherwise our program would have considered them misspelled Italian words and wrongly counted them as errors. The task proved to be difficult as there is no comprehensive English food terms' corpus publicly available. To solve the

[2] Refer to Appendix B for results from the sample test menus.

problem we created an "English words filter" by creating our own corpus and by combining it with the nltk English words corpus and using stemming and lemmatization.

- **Lack of context.** As we only worked on dishes and ingredients out of the context of a sentence, more sophisticated techniques for grammar checking and similarity measurement (e.g. word embeddings and n-grams models) were not applicable.
- **Regional/dialectal words.** As anticipated, the program labels as errors some correct words that are dialectal words or that are only used in certain Italian regions.
- **N-grams detection.** Given the menus composition of mere dishes, it was complicated to extract usable n-grams in a consistent way, especially for test samples where we could not create proper tables of n-grams per term as we could do for our main corpus.

6. Future Work

This project could be expanded in two main directions:

- The improvement of the typos detection system, which we have prototyped here. This will be only doable by: 1) extending our own background corpus to include many more diverse terms as well as counting for dialects and cultural assimilation; 2) improving the reading/cleaning of test menus to get the best possible set of terms to be assessed.
- A parallel linguistic analysis could also be attempted, by looking for example at hash tables with closest neighbors of terms (i.e. rough n-grams) and compare the use of prepositions and gender agreement, even though proper grammatical checks will not be feasible and these approaches seem much more approximate than desired in our case. An interesting use of word embeddings could be applied for a more nuanced scoring system as we could weigh terms by their significance within a menu.

Appendix A

Scoring system:

Mistakes	Score	Label	Image
0-5	5	IGA (Italian-Grandma-Approved)	
6-10	4	Second Generation	
11-15	3	Local Italian Restaurant in Toledo, OH	
16-20	2	Olive Garden	
20-25	1	Deep Dish Pizzeria	

Appendix B

Example of application on sample menus:

	Restaurant	Errors	# Errors	Score
5	Il Canale - Washington, DC	[georgetown]	1	5
7	Osteria la Spiga - Seattle, WA	[ragu, ida, stridoli]	3	5
1	La Storia - Chicago, IL	[neri, orecchitte, agnelo, malloreddus, sautee]	5	5
8	Via dei Tribunali - Seattle, WA	[cappuccino, caffe, dante, tribunali, campania...	9	4
4	I Ricchi - Washington, DC	[salsiccie, buratta, paillar, costoletta, tort...	10	4
3	Otello - Washington, DC	[cappuccino, episelli, mirella, capellini, sea...	11	3
9	Il Terrazzo Carmine - Seattle, WA	[affogati, spaghettini, asparaci, animelle, so...	13	3
6	Valentinos - Nashville, TN	[arugula, parmesan, manhattan, scottish, scarp...	16	2
0	Ristorante Piccolo - Washington, DC	[arugula, cioppino, altopiano, scarpariello, i...	17	2
2	Portofino - Arlington, VA	[vinci, omaggi, picatta, cioppino, almondine, ...	18	2

The code and result of this sample application of our program can be found at the following notebook: https://github.com/matteobucalossi50/FIRe-Fake-Italian-Restaurants-evaluator/blob/master/FIRe-example.ipynb

User-based application:

To check the authenticity of a restaurant of your choice, you can download the repository and run 'python fire.py' in the terminal within our repo. Use exclusively urls from opentable.com when asked to provide one, and the program will label the "Italianity" of the menu for you!

References:

- GialloZafferano: https://www.giallozafferano.it/
- Gonen, Ila et al. "How Does Grammatical Gender Affect Noun Representations in Gender-Marking Languages?" Conference paper presented at 23rd Conference on Computational Natural Language Learning, January 2019, retrieved from https://www.aclweb.org/anthology/K19-1043.pdf
- Jurafsky, Daniel and Martin, James. 2009. *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd edition. Prentice-Hall.
- Konchady, Manu, "Detecting Grammatical Errors in Text using a Ngram-based Ruleset", January 2009, retrieved from https://www.researchgate.net/publication/255654796_Detecting_Grammatical_Errors_in_Text_using_a_Ngram-based_Ruleset
- OpenTable: https://www.opentable.com/
- PyEnchant: https://pyenchant.github.io/pyenchant/
- Spell Checker with Word2Vec:Build a spell-checker with word2vec data (with python)
- SymSpell: https://github.com/wolfgarbe/SymSpell
- Tabula: https://tabula.technology/