MASTER'S IN COMPUTER SCIENCE
ARTIFICIAL INTELLIGENCE

EMOJISLATION

TRAITEMENT AUTOMATIQUE DU TEXTE EN IA

Charafeddine Achir & Rafael Baptista
Université Côte d'Azur
2024

TRAITEMENT AUTOMATIQUE DU TEXTE EN IA

FINAL PROJECT REPORT

Subject: Emojislation

Group : Charafeddine Achir and Rafael Baptista

Academic Advisor : Elena Cabrio

Abstract :

The Emojislation project is focused on translating English phrases into corresponding sequences of emojis. This translation process is achieved through the application of vector cosine similarity, which identifies the most suitable emoji for each word by comparing word vectors with emoji vectors. Central to the project are two key resources: 'emo_uni', a library that provides textual representations for each emoji, and the 'emoji' Python library, enabling the conversion of words to emojis and vice versa. Additionally, the project uses GloVe (Global Vectors for Word Representation), an extensive database of word vectors, integrated with spaCy, a tool for natural language processing, to generate corresponding emoji vectors. This endeavor provides a comprehensive exploration into language processing, bridging the gap between textual and visual communication and enhancing the understanding of natural language processing techniques.

Summary :

1. Context

Our project, entitled "Emojislation", was conceptualized and developed as part of our “Traitement automatique du text en IA” (TATIA) class. The primary objective of the class was to explore innovative applications of NLP and to gain hands-on experience in implementing these applications in real-world scenarios.

Our project emerged from two pivotal concepts: a desire to delve into practical applications of natural language processing and an interest in merging the expressiveness of emojis with textual communication. Motivated by the growing prevalence of emojis in enhancing digital conversations, our objective is to develop a system that translates English sentences into emojis. This initiative not only reflects modern communication trends but also showcases the potential of language processing in creating dynamic, visually enriched interactions online.

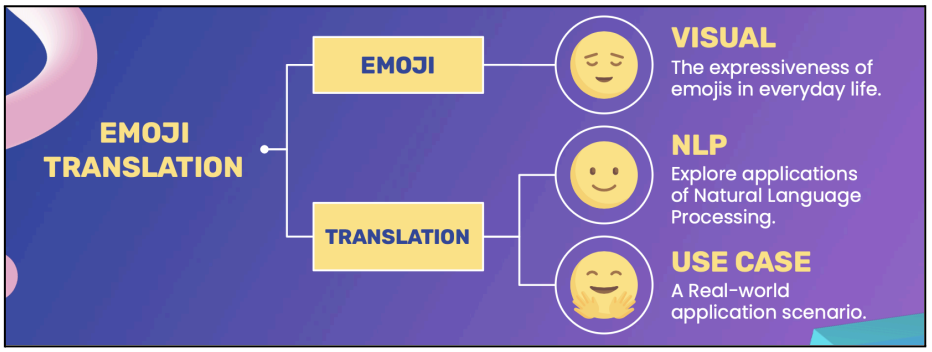


Image 1 : The brainstorming context

This project was more than just a class assignment. It was a chance to explore the fun side of NLP and see how far we could push the boundaries of digital communication. Emojislation was our playground – a place where we could mix language and technology, to see what we could create.

2. Introduction

Emojislation aims to harness the universal appeal of emojis, translating English text into these expressive symbols. This initiative goes beyond traditional text analysis, delving into how emojis can represent textual meaning in online communication. Leveraging the power of NLP tools like GloVe for word representation and spaCy for processing, our project goes beyond mere word-to-symbol translation.

This is an experiment in understanding digital language, exploring how emojis can encapsulate emotions and ideas that words alone might not fully convey. This report will detail the journey of Emojislation, from conception to implementation, shedding light on the challenges we faced and the insights we gained in blending the realms of text and emojis.

3. Process description

In this section we will describe the technical process of Emojislation involving several key steps, each integrating NLP techniques and tools.

3.1. Generating a textual dataset

Since we required a reliable textual source for testing the project, it was crucial to provide a diverse and extensive set of simple English phrases. Using GPT-4, we generated a total of 400 phrases. The focus was on creating simple, generic English phrases, like those typically found in language textbooks.

```
    "They are playing soccer.",
    "I am eating an apple.",
    "She is drinking water.",
    "The baby is crying.",
    "We are watching a movie.",
    "The birds are singing.",
    "He is running fast.",
    "The door is closed.",
    "She is cooking dinner.",
    "The phone is ringing.",
    "They are dancing together.",
    "I am learning English.",
    "The car is moving.",
    "She is wearing a dress.",
    "The flowers are blooming.",
    "He is swimming in the pool.",
    "They are riding bicycles.",
    "I am drawing a picture.",
    "The train is arriving.",
    "She is playing the piano.",
```

Image 2 : Sample of generated english sentences with GPT-4

3.2. Processing Emoji icons

In our project, we leveraged the Python Emoji library to effectively display and print emojis using their associated keywords. This library simplifies the process of handling emojis in Python, allowing us to map textual descriptions or keywords to their corresponding emoji symbols. By utilizing this tool, we were able to seamlessly integrate emojis into our translation system.

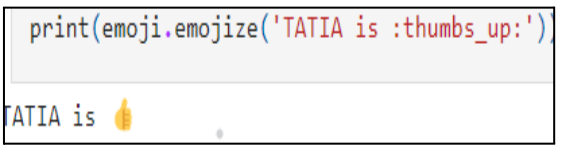


Image 3 : Python's Emoji library

3.3. Textual Representation for Emoji

Our approach began with processing of emoji data, which was essential for translating English text into emojis. We used a comprehensive list of emojis, as defined in the emo_uni.py (the annex file can also be found in the project's github page in the directory :

https://github.com/rafaelbenaion/emojislation/blob/main/emo_uni.py) file, which served as our primary emoji library. This file provided a textual representation for each emoji, mapping emoji names to their corresponding Unicode characters.

4. The translation process

We are going to explore in this section all the necessary tasks achieved for the translation process in place.

4.1. The SpaCy

We have used SpaCy, an advanced language processing tool, to analyze text. We started with the 'en_core_web_sm' model from SpaCy, which is great for dealing with English. Using SpaCy was an important step. It helped us understand and interpret our text in detail, setting us up for accurate translations later on.

4.2. The Glove

For our word embeddings, we chose GloVe, trained on a Wikipedia dataset. We loaded a 300-dimensional version using a standard file read method. By integrating these vectors into SpaCy's vocabulary, we could represent words in a way that captured their meanings deeply. This was crucial for translating text into emojis while keeping the original intent of the words.

4.3. Cosine Similarity

We relied on cosine similarity to find the most suitable emojis for our text. By developing a function to compare word vectors with emoji vectors, we could pick the emoji that best matched each word. This approach was key to ensuring our translations were accurate and meaningful.

4.4. The Threshold

Setting a threshold was an important part of our system. We decided on a minimum cosine similarity score of *0.0115*. This meant we only chose emojis that closely matched our text, ensuring our translations were both relevant and appropriate. It was a simple yet effective way to keep our emoji translations on point.

5. Validation Study Case

One of the biggest challenges in our project was validating our model. The unique nature of our approach to emoji translation meant that we couldn't rely on traditional methods typically used in machine learning, like splitting data into training, validation, and test sets. This deviation from convention necessitated innovative thinking and a creative solution for testing the efficacy of our model. We recognized early on that the validation process needed to be as unconventional and forward-thinking as the project itself. Therefore, we endeavored to devise a validation strategy that was not only effective but also aligned with the distinctive characteristics of our translation tool. This led us to develop an interactive and user-centric method of validation, ensuring that our approach was thoroughly tested and refined based on real-world feedback.

5.1. The Platform

To validate our emoji translation project, we developed a unique approach. We created a web interface showcasing 300 phrases alongside their corresponding emoji translations. Each phrase-emoji pair was accompanied by a small checkbox as shown in the Image 4. We then distributed this interface to a broad audience, inviting them to participate in our study. Participants were asked to tick the boxes next to the translations they felt accurately matched the given phrases. This interactive method not only engaged the users but also provided us with valuable data to gauge the effectiveness of our translation model. The feedback from this exercise resulted in an accuracy score of 58% for

our model. Considering the subjective and interpretative nature of emojis, this score is quite commendable. It highlights the practical effectiveness of our translation system and provides a solid foundation for further improvements.

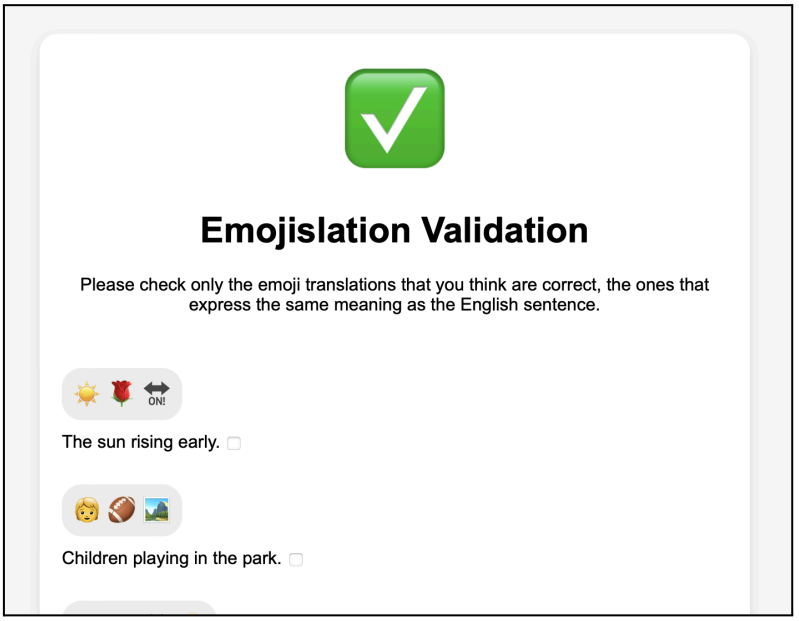


Image 4 : The Validation Platform

5.2. The Dashboard

To further enhance our project's analysis and presentation, we directly integrated our collected data into a MongoDB database. This step allowed us to systematically store and manage the feedback received from our web interface validation. Using MongoDB's robust capabilities, we created a comprehensive dashboard, as shown in Image 5. This dashboard vividly presents all the essential information, offering an intuitive and interactive way to visualize the data. It not only showcases the accuracy score but also provides deeper insights into user responses and trends. This integration has been instrumental in allowing us to efficiently analyze the results and draw meaningful conclusions, further validating the effectiveness of the translation model.

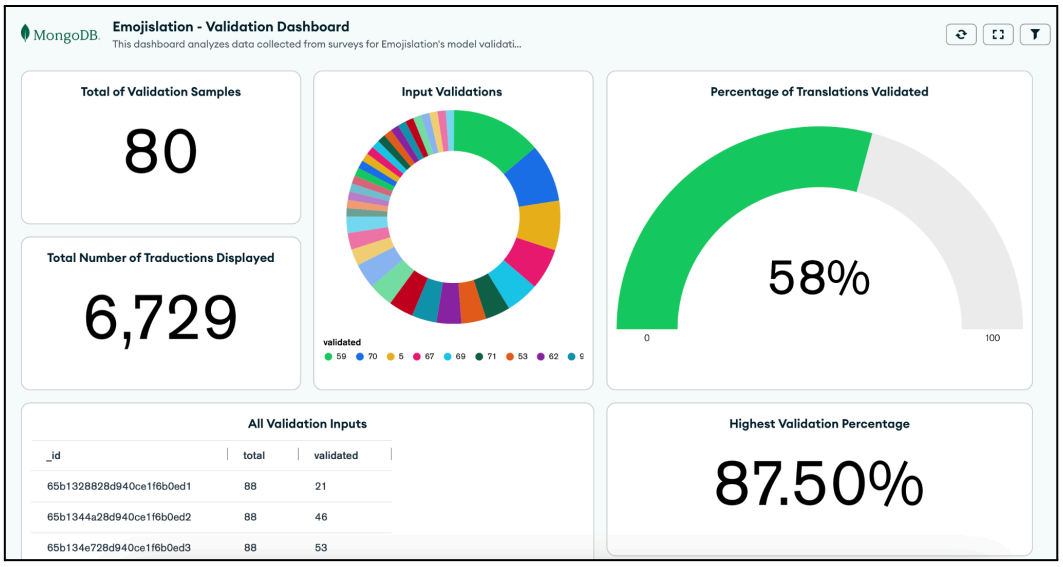


Image 5 : The Validation Dashboard

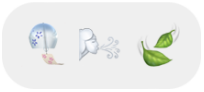
6. Conclusion

In conclusion, our project embarked on the ambitious task of translating text into emojis, navigating through unique challenges and learning valuable lessons. Initially, we faced a significant hurdle: our system was constrained to match the number of words in a sentence with an equal number of emojis. This one-to-one approach, however, didn't always yield meaningful translations, as not all words have direct emoji counterparts. To address this, we introduced a threshold for similarity scores. By doing so, we ensured that only words with a high degree of relevance and similarity to an emoji were translated, enhancing the overall quality and coherence of our translations.

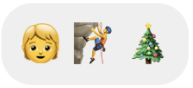
Another major challenge we encountered was with articles and to-be verbs. Our system struggled to find suitable emojis for these, as they often lack direct visual representations. This led us to an important decision: simplifying the sentences by removing these elements. This approach not only solved the immediate problem but also streamlined our translations, making them more focused and impactful.

Through these adjustments, our project evolved from a rigid, word-for-word translation model to a more nuanced and selective approach,

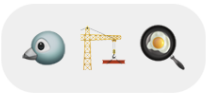
prioritizing relevance and clarity. These experiences highlight the iterative nature of problem-solving in computational linguistics and the importance of adaptability in the face of unforeseen challenges. Our journey through this project not only advanced our technical skills but also deepened our understanding of the complex interplay between language and symbolism.



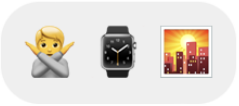
The wind blowing the leaves.



The kids climbing the tree.



Birds building nest.



They watching the sunset.

The clock ticking loudly.

The sun setting.

7. Useful Links

Here you can find all the links for the GitHub repository that contains all the files used in the project for making it turn locally. You can also find the link for the validation dashboard that contains all data collected during the validation phase through the surveys, and the validation platform.

7.1. GitHub project - Emojislation :

https://github.com/rafaelbenaion/emojislation

7.2. Atlas MongoDb - Validation Dashboard :

https://charts.mongodb.com/charts-project-0-uqrbc/public/dashboards/65b1b664-d896-436d-8e7d-f80f0b8bcad3

7.3. The web platform :

https://emojislation.onrender.com