Sentence Compression Using Emoji Summarization

Alex Day and Chris Mankos

Faculty Advisors: Dr. Soo Kim and Dr. Jody Strausser
Computer Information Science Department at Clarion University

Overview

- Introduction
  - Problem Statement
  - Importance of Emoji Summarization
  - Related Work
- Composition of N-grams for Emoji Translation Algorithm
  - Sentence Composing
  - N-Gram → Emoji
  - Translation Scoring
  - Sentence Generation
- Results
  - Results from our algorithm
  - Future work

Problem Statement

- Research Question: Can we effectively summarize sentences using emojis using vector embeddings to produce meaningful results?
- Not a complete one-to-one mapping of words to emojis
  - Capture ‘essence’ of several words
- Some examples of what we would like:
  - My dog can run so fast → 🐕🏃💨
  - I’m thinking that this computer has a virus → 🤔🖥️🦠

Importance of an Emoji Summary

- Abstract
  - Translation will always improve machine understanding
  - Improve emoji understanding
- Concrete
  - Facilitate in communication across language barriers

Related Work

- Embeddings
  - Word2Vec
  - Sent2Vec
  - Emoji2Vec
- Direct word → emoji mappings
  - https://decodeemoji.com
  - https://meowni.ca/emoji-translate/
- Emoji Dick
  - Translation of Moby Dick into Emojis
- Text Summarization

Composition of N-Grams for Emoji Translation (CoNET)

- CoNET is a combination of machine learning and natural language processing (NLP) techniques that can produce a series of emojis when given a variable length input sentence
- CoNET does **not** accomplish this with a lookup table for keywords relating directly to emojis
- Algorithm is split into separate parts
  1. Sentence compositions
  2. Part → emoji comparison
  3. Summary scoring
  4. Summary generation

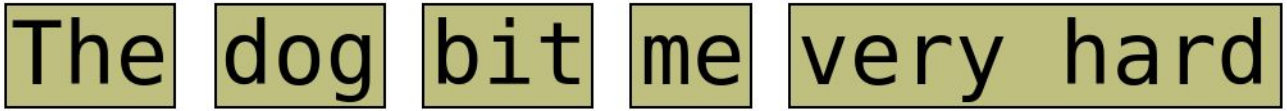
Visual for overall flow

Sentence Composing

- An n-gram is a variable length sequence of contiguous words, normally in the context of a larger phrase or sentence.
  - A sentence can be represented by a sequence of n-grams
  - **Ex:** The sentence “The dog bit me very hard” has the n-grams:
    - “The dog bit”, “me”, “very hard”
- We will refer to a sequence of n-grams as the **n-gram sequence**, and an individual n-gram in the sequence as an **n-gram**

Sentence Composing Continued...

- The simplest way to partition a sentence is to do so exhaustively
  - **Ex:** For the sentence “The dog bit me very hard” we check all sequences of n-grams:



- Assumption: there must exist some optimal n-gram sequence that generates the best summary

N-Gram→ Emoji Comparison

- Need a way to translate an n-gram (eg. “the dog”) to an emoji (eg. 🐕) that is **not** just a one-to-one mapping from word to emoji
- The Embedding of a phrase, word, or emoji is a point in space that represents the meaning of that phrase, word, or emoji. The space is normally between 300 and 700 dimensions of numbers between 0 and 1.
    - vec(King) - vec(Man) + vec(Woman) = vec(Queen)
- We can calculate the similarity between an emoji and an n-gram by calculating the cosine difference between the emoji’s description and the n-gram

N-Gram→ Emoji Comparison Continued...

- The dataset we are using for emojis contains a series of emojis and then multiple descriptions for them
- Cosine Similarity is the cosine of the angle between two points with respect to the origin
- We can calculate the closest emoji to an n-gram by calculating the cosine distance between the n-gram and the emoji description and returning the emoji with the highest similarity

Translation Scoring

- We score a sentence based on the sum of its parts. Meaning that the sentence's score as a whole is an average of the cosine similarity of the n-gram → emoji pairs that make up that summary.
- 
  - N-grams → "the dog" "runs" "fast"
  - Emoji-grams → "dog" "run" "fast"
  - Cosine Similarity → 0.96, 1.0, 1.0
  - Average Cosine Similarity → 0.9844
- 
  - N_grams → "i think that this" "computer" "has a virus"
  - Emoji-grams → "think" "computer" "virus"
  - Cosine Similarity → 0.52, 1.0, 0.79
  - Average Cosine Similarity → 0.231

Summary Generation Algorithm

1. Given a sentence, S, to summarize
2. Split S into every possible n-gram sequence, call that list of sequences N
3. For every sequence in N:
    a. For every n-gram in sequence
        i. Find closest emoji and add that to the summary
    b. Score sequence
4. Return sequence in N with highest score

Results

Input Sentence	Output Emojis	Score
The dog runs fast		0.984
The child was in love with the cat		0.824
They are playing christmas music from the bell tower		0.893
I think that this computer has a virus		0.769
I have to wear my headphones to run in the race		0.960
The company Apple makes both cell phones and computers		0.903

Future Work

- Improved dataset
  - The dataset is the main influence on the “readability” of the generated summaries.
  - Experimentation with Emojipedia
- Each n-gram is currently independent of every other n-gram in the sequence
  - By checking before and ahead and using that to influence the decision it may lead to better results. This is a proven technique used by Recurrent Neural Networks.
- Improve chunking
  - Generate chunks with information about sentence composition
  - Score chunks before they are summarized
- Improve Testing Metrics
- Sentiment emoji

Questions?