I'm working with a CSV file of 2-gram phrases from news articles spanning several months. I filtered the top 50 most frequent 2-grams, but they don’t seem to show any clear pattern or meaningful insight. I'm trying to understand how to analyze this data in a more insightful way, especially to detect patterns in tone, framing, across time. Can you help me think about better ways to categorize or group these bigrams so I can compare meaningful trends? Don’t tell me the code, just act as a tutor

2/2

**ChatGPT said:**

Absolutely—let’s approach this like a tutor would: by thinking together about *how* you can extract meaning from those bigrams.

**Step 1: Think Beyond Raw Frequency**

Frequency alone doesn't tell you much about *context*, *tone*, or *change over time*. So let’s ask:

* What do these bigrams *represent*?
* Can they be grouped by *themes*, *sentiment*, or *frames*?

**Step 2: Categorise the Bigrams Thematically**

Ask yourself:

* Are these bigrams about **conflict** (e.g. “air strike”, “military action”)?
* Are they about **politics**, **humanitarian issues**, **international relations**, **diplomacy**, etc.?

You can try manually labelling the top 50 into rough **semantic categories**. Even 3–5 broad labels can reveal patterns over time.

Examples:

* **Violence/Conflict**: “missile attack”, “gun fire”
* **Humanitarian**: “food aid”, “civilian casualties”
* **Diplomatic**: “peace talks”, “cease fire”

**Step 3: Map Tone or Emotion**

Look at the *valence* of bigrams:

* Are they **positive**, **negative**, or **neutral**?
* Try using **emotion or tone lexicons** (like NRC Emotion Lexicon or LIWC categories) to associate a tone with each word in the bigram.

You can then:

* Assign scores (manually or lexicon-based) to see shifts over time.
* Group bigrams by emotions: fear, anger, sadness, hope, etc.

**Step 4: Use Time Windows to Compare**

Create a timeline—maybe by week or month—and see:

* Which category (from above) is most dominant in each window?
* Does the framing shift? For instance, do we move from “military offensive” to “peace deal”?
* Do emotions rise or fall in correlation with certain events?

This temporal comparison often reveals narrative shifts.

**Step 5: Try Co-occurrence and Contextual Proximity**

Ask:

* Which bigrams often appear *near each other*?
* Can you cluster them based on articles or topics?

This helps you move from isolated bigrams to **narrative clusters**.

**Final Thought: Frame and Reframe**

Think like a media analyst or anthropologist:

* What *story* is being told through these 2-grams?
* Can you connect the tone and topic categories to known **frames** like “victimhood”, “aggression”, “resistance”, “legitimacy”?

Would you like help building a set of categories together or thinking through a test case from your list?

**You said:**

. I ended up grouping the 2-grams into “Sympathy” and “Aggression” themes and then tracked how often they appeared each month. It showed clearer trends than just frequency.

**ChatGPT said:**

That’s excellent! Grouping into **Sympathy** and **Aggression** is a strong, conceptually meaningful divide—perfect for highlighting shifts in tone or framing.

Here are some ideas to deepen this approach further:

**1. Refine Your Themes**

* Can you add **neutral** or **other** categories for bigrams that don’t clearly fit sympathy or aggression?
* Maybe break them down further, like “Sympathy → Humanitarian Aid vs Civilian Suffering” or “Aggression → Military Action vs Political Threats.”

**2. Look for Leading/Lagging Patterns**

* Does an increase in Aggression bigrams precede a drop or rise in Sympathy bigrams?
* Can you link these shifts to real-world events or media cycles?

**3. Consider Intensity or Strength**

* Are all aggression bigrams equally intense? (“bomb attack” vs “armed forces” might carry different emotional weights)
* You could weight bigrams by intensity or emotional impact.

**4. Cross-Reference with Other Data**

* Does sentiment from full articles or headlines correlate with your bigram categories?
* What about other metadata like source, region, or article type?

**5. Visualisation**

* Plot your Sympathy and Aggression counts as stacked bars or lines over time to easily spot trend shifts.
* You could also visualize the ratio of Sympathy to Aggression for each month.

Would you want to explore ways to test or validate your theme assignments, or how to connect these insights to broader narratives or discourse analysis?

**You said:**

How can I make sure that the 2-grams I’m analysing are meaningful and don’t include common stopwords or invalid words I used stop words but still some are showing?

**ChatGPT said:**

Great question — ensuring your 2-grams are meaningful is key to a solid analysis! Here’s a step-by-step approach you can follow, conceptually:

**1. Remove Stop words Before or After Extraction**

* Stopwords are very common words (like *the, and, of*) that carry little semantic meaning.
* Usually, you **filter stopwords before building bigrams**, so pairs like “of the” don’t appear.
* If you’re working with bigrams already extracted, **remove any bigram where either word is a stopword**.

**2. Filter Out Non-Informative Phrases**

* Some 2-grams might be frequent but not meaningful (e.g., “said that,” “according to”).
* You can maintain a **custom list of ‘bad’ or ‘invalid’ bigrams** to exclude.
* Or manually skim the top N to identify and remove such cases.

**3. Check Parts of Speech (POS)**

* Use POS tags to keep only bigrams with meaningful word types, like:
  + Adjective + Noun (“military action”)
  + Noun + Noun (“peace talks”)
  + Verb + Noun (“launch attack”)
* Avoid bigrams made only of function words or unlikely pairs (e.g., preposition + preposition).

**4. Frequency Thresholding**

* Remove bigrams that appear very rarely—these may be noise.
* Also consider removing excessively frequent but generic bigrams.

**5. Contextual Validation**

* For borderline cases, check sample articles where the bigram occurs to see if it really adds meaning.
* This helps avoid including misleading or ambiguous bigrams.

**Bonus Tip:**

* Sometimes **lemmatisation or stemming** helps unify variants (e.g., “attack” and “attacks”) and reduces noise.

Does this help? Want to brainstorm specific stopword lists or POS patterns to filter?

**You said:**

teach me How can I label or classify each 2-gram in my data as belonging to a specific theme like “Sympathy” or “Aggression” based on predefined sets of phrases?

**ChatGPT said:**

Sure! Here’s a step-by-step way to *conceptually* label or classify each 2-gram into your themes like **Sympathy** or **Aggression**, based on predefined phrase sets:

**Step 1: Define Your Theme Phrase Lists**

* Prepare **two lists (or dictionaries)**: one for Sympathy bigrams, one for Aggression bigrams.
* For example:
  + Sympathy: “civilian casualties”, “humanitarian aid”, “refugee crisis”
  + Aggression: “military attack”, “armed forces”, “bomb explosion”

**Step 2: Normalize Your Data**

* Before matching, convert all bigrams to lowercase to avoid case mismatches.
* Optionally, strip punctuation or extra spaces.

**Step 3: Exact Matching**

* For each bigram in your dataset, check if it appears in your Sympathy list or Aggression list.
* If yes, label it accordingly.
* If it’s in neither, you can label it as **Unclassified** or **Other**.

**Step 4: Handling Ambiguities or Overlaps**

* If a bigram appears in both lists (rare but possible), decide on a rule:
  + Assign priority to one theme, or
  + Mark as **Ambiguous** for manual review.

**Step 5: Expand Your Lists Over Time**

* Start with a small core set of bigrams per theme.
* As you review data, add new relevant bigrams to your lists to improve coverage.

**Step 6: Optional — Use Partial or Fuzzy Matching**

* If you want to be flexible, you can check if a bigram contains keywords from your theme lists (e.g., any bigram containing “aid” → Sympathy).
* But be cautious: this can introduce noise.

**Step 7: Validate Your Labels**

* Spot-check a sample of labeled bigrams with their original context to ensure accuracy.
* Adjust your lists or rules accordingly.

Would you like to discuss how to build those phrase lists, or how to handle bigger datasets with this approach?

**You said:**

need further explanation in simple words

**ChatGPT said:**

Got it! Let me explain it in a simple way:

**How to label each 2-gram as "Sympathy" or "Aggression":**

1. **Make two lists of phrases you know fit each theme**
   * For **Sympathy**, write down phrases like "humanitarian aid", "civilian casualties"
   * For **Aggression**, write down phrases like "military attack", "bomb explosion"
2. **Clean your 2-grams**
   * Change all the words to lowercase so everything looks the same (like "Humanitarian Aid" → "humanitarian aid")
3. **Look at each 2-gram in your data**
   * Check if it matches a phrase in your Sympathy list → if yes, label it "Sympathy"
   * Check if it matches a phrase in your Aggression list → if yes, label it "Aggression"
   * If it doesn’t match any list, label it "Other" or "Unclassified"
4. **If a phrase shows up in both lists**
   * Decide which label is more important for you, or mark it to review later
5. **Keep adding new phrases**
   * When you find new important 2-grams, add them to your lists to improve your labeling
6. **Check your work**
   * Pick some labeled 2-grams and look at the original articles or sentences to see if your labels make sense