Novel Generative Summaries: Safer Prompts & More insightful Market Research?

Finn-Ole Höner (657110)

June 21, 2024

Introduction

Many marketing problems in NLP require an accurate representation of the information contained in a text document. For example, we might want to analyze the design elements of advertising claims and predict their perception by consumers, or analyze the sentiment of a product review (Berger et al. 2020). A powerful advance in NLP, are Large Language Models (LLM), such as GPT (Radford et al. 2018) and the demand of businesses for such tools is growing rapidly[1]. In marketing, summarizing texts with these LLMs is becoming an important application (Hartmann and Netzer 2023). However, using an LLM to summarize a document is difficult: The generated summaries depend on the wording of the used prompt and can be unreliable. Such textual summaries are also difficult to evaluate and are not practical to use in prevalent data science applications, such as prediction or clustering. In this work, we show how we can summarize documents with LLMs for this purpose.

A core distinction of our approach is, that we use an LLM to summarize a document without relying on prompt engineering. Prompt engineering defines the task of finding the right prompt to generate a desired output with a LLM. Typically, this is an interactive and iterative process between humans and LLM. Certain writing techniques improve the quality of the LLM's response, such as providing examples of what the desired output should look like. There are various blog posts[2] and books (e.g. Phoenix and Taylor 2024) about how to write good prompts. Businesses are hiring "Prompt Engineers"[3] on dedicated job boards[4], indicating a demand for operating LLMs in a better way. Despite these, supposedly low-barrier, resources, end users without AI knowledge still struggle with prompt engineering (Zamfirescu-Pereira et al. 2023). Not only can prompt engineering to pose a security risk[5], but the process is also fuzzy, hard to replicate, and might still lead to unpredictable behaviors of the LLM. These issues pose risks for organizations, as the obscurity of manual prompt engineering can lead to loss of know-how, and the unpredictability of the model can lead to lawsuits, damages to brands, and missed opportunities. Problems of reproducibility, transparency, and documentation become critical for organizations with the advent of the EU AI Act[6]. The act demands that the AI model features "adequate mitigation systems", "traceability of results", and "robustness" when used for "High-risk applications". High-risk applications are common and important for society, as they include applications in education, employment, and public services. Hence, businesses and society alike have an interest in improving prompt engineering, in a way that makes the responses of LLMs better in quality and more robust.

To make prompt engineering more safe, we need to make an LLM's output more reliable. In this study, we make a step towards more reliable text generation by proposing a novel type of document summary, which

[1] https://www.statista.com/outlook/tmo/artificial-intelligence/natural-language-processing/worldwide#market-size
[2] https://huggingface.co/docs/transformers/en/tasks/prompting#best-practices-of-llm-prompting
[3] https://indeed.com/q-prompt-engineer-jobs.html?vjk=98ac28acfd1328d7
[4] https://prompt-engineering-jobs.com/jobs/prompt-engineer-qwvmju/
[5] https://www.promptingguide.ai/risks/adversarial
[6] https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

we call the "generative summary". The core idea of our method is to find an input to the LLM that captures all the information contained in a focal document, as it leads the LLM to recreate this document itself. To obtain this summary, we maximize the likelihood of generating the document with a Large Language Model (LLM) given a numeric vector. Hence, these generative summaries are optimal. To mitigate overfitting and to make these embeddings more interpretable, we also introduce a factor model to estimate these document summaries in a lower dimensional space. We show that these embeddings capture inherent information about the document, and can be used for the generation of new documents. We also find that we can form linear combinations of generative summaries that have meaning themselves, similar to the approach by Mikolov, Ilya Sutskever, et al. (2013) with word embeddings.

Furthermore, we illustrate the use of our generative summaries in a marketing setting and show how they can yield insights in market research. We apply our method to two datasets. First, an artificial dataset of advertising claims for hair shampoo and surface cleaners to validate that these generative summaries capture relevant information. Second, we show an application of these generative summaries in market research. We use these generative summaries to analyze market research data on product claims for a yogurt and a yogurt drink. We find that we can use these embeddings to capture the design elements of these claims, assess their linguistic uniqueness, and represent their design patterns. Surprisingly, we find that this linguistic uniqueness is negatively correlated with the perceived uniqueness of an advertising claim by consumers. We explore these linguistic features and find that claims with certain words and their synonyms cluster together, e.g. claims that emphasize the taste of a yogurt drink or present it as a breakfast. Furthermore, we show that claims that live in certain areas of a low dimensional representation of these summaries are more likely to be rated favorably by consumers, but that small differences in wording can lead to relatively large differences in evaluation. This suggests a two-step approach to designing a market research study for advertising claims, where the first stage should focus on a wide exploration of the design space, while the second stage should focus on a fine-grained evaluation of the most promising area in this design space. We end with a sketch of how managers can use these generative summaries to augment the design process and market research on product claims.

We proceed with a brief overview of the relevant literature and introduce our methodology. After testing our procedure on synthetic data, and exploring the low-dimensional factor space for the generation of new advertising claims, we apply our model to the market research data. We conclude with managerial implications, a discussion of the limitations of our approach and expansions for this research.

Relevant Literature

In Natural Language Processing, words, symbols, and even syllables are represented by so-called tokens, integer codes representing a sequence of characters. A piece of text, represented by tokens, is what we call a document. To represent the meaning of words and phrases, rather than to just encode them by a token, we can use word embeddings. Word embeddings are numeric vectors that represent text. They are pre-trained on large amounts of text data and can capture the meaning of single words and phrases. Early approaches include the Word2Vec model by Mikolov, Ilya Sutskever, et al. (2013). The researchers train these word embeddings through a model that predicts the text surrounding a focal word. Thereby they learn the context in which words occur, which is a way of representing their meaning. There are also approaches that create embeddings for sequences of words or even documents (e.g. Devlin et al. 2018). In this work, we use Large Language Models (LLMs). LLMs, such as ChatGPT[7] (Radford et al. 2018), predict the next word in a sequence of words. These models generate text based on textual inputs ("prompts"). Internally, the model translates these prompts into a sequence of vectors, the "input embedding". This input embedding is then passed through a neural network that predicts the next token in the sequence. To generate text, the model predicts the next token and attaches it to the prompt, repeating this process until it predicts the sequence to end (i.e. predicting the next token to be the "End-of-Sequence" token: "eos-token"). LLMs are pre-trained on large text corpora, such as the Common Crawl[8] which contains over 250 billion pages of text. Their architecture is rooted in the attention mechanism. The attention mechanism

[7] https://chat.openai.com/
[8] https://commoncrawl.org/

models interactions in the text, such as negations, synonyms, or grammatical structures, as well as topics, across a long sequence of text. Vaswani et al. (2017) implements this mechanism in the transformer block, which is a modular building block for the LLM. The combination of these transformer blocks, giant pre-training data, and huge computing resources lead to LLMs that have "emergent capabilities". These are capabilities, that were not explicitly trained for by the developers of the model, such as the ability of a model to translate between languages, pass the BAR exam, or write textual summaries of documents (see Wei et al., n.d.; Lu et al. 2023).

In the following, we compare our *generative summaries* (*GS*) with three established methods to summarize documents:

- The Bidirectional Encoder Representations from Transformers (*BERT*) model's classifier token (CLS) (Devlin et al. 2018)
- Pooled word embeddings (*PWE*) (Shen et al. 2018)
- Prompt engineering approaches (*PE*) (Huang and Chen, n.d.; Khattab et al. 2023)

BERT: An advanced approach to estimate a word embedding is the *BERT* model, which is a transformer-based model to represent text (see Vaswani et al. 2017). Its developers have pre-trained BERT on a large corpus of text, using two different self-supervised learning objectives. The first objective is the so-called "Masked Language Model" task. Here, the researchers hide a random token in a sequence, and the goal of the BERT model is to predict which token is missing. This teaches the model the meaning of words in context. To learn about the information captured in a sentence, the researchers also pose the "Next Sentence Prediction" task to the model. For this training objective, the BERT model receives a pair of two sentences and needs to predict whether the second sentence follows the first one in a text. This task trains the classifier token (CLS): As it is pre-pended before every such sentence pair, it learns a representation of the information contained in the focal sentence. In practice, the output for a CLS token serves as a powerful feature e.g. for sentence classification. BERT embeddings are versatile, but can also be fine-tuned to a specific task such as text summarization. Such a fine-tuning requires the user to compile a dataset of summary-text pairs, that are representative of their application. These BERT representations are only descriptive and cannot be used for the generation of new text (see Devlin et al. 2018). In contrast to generative LLMs, such as GPT (Radford et al. 2018), which consider context only unidirectional, BERT takes text from the left and the right into account when representing it as an embedding. There exist improvements to BERT, such as RoBERTa by Yinhan Liu et al. (2019), which improve the optimization procedure.

PWE: Shen et al. (2018) argues that pooled word embeddings (*PWE*) are a simple, yet powerful way to represent documents. The idea behind a *PWE* is that we can represent a sequence of tokens by aggregating the word embeddings of these tokens. There are different methods we can use for this aggregation, such as taking the average across the tokens (mean-pooling) or taking the maximum value across the tokens (max-pooling). There are various types of word embeddings that we could use for the pooling, such as Word2Vec (Mikolov, Ilya Sutskever, et al. 2013; Mikolov, Kai Chen, et al. 2013) or Global Vectors (GLoVe) (Jeffrey Pennington et al. 2014). These pooled word embeddings, cannot be used for text generation and perform worse than BERT embeddings in language tasks, as they do not employ the attention mechanism and lose information in the aggregation step (see e.g. Onan 2023).

PE: We can also summarize documents by passing these to an LLM and prompting the model to write a summary for us (Prompt engineering approaches, *PE*) (e.g. Huang and Chen, n.d.; Chakraborty and Pakray 2024). However, these summaries are textual, not deterministic, and their quality depends on the formulation of the prompt. For example, LLMs deliver better answers when the prompt contains relevant information at the beginning or the end, or when the prompt is written emotionally (see Liu et al. 2023; Li et al. 2023). Another issue is that these summaries can vary in length, and one needs to specify how detailed a summary should be. The difference in length between the summary and focal document, also makes it challenging to evaluate how much of the information of the focal document is captured by the summary (Chakraborty and Pakray 2024). If these summaries are of a high quality, i.e. represent the information in the focal document well, then they can be useful when we want to share a written summary of a document.

In this paper, we propose a solution that works for data science and automation tasks, as we create a numeric representation of the document. We could use these textual summaries for generation, in the sense that we could prompt the model to generate a new document based on the summary.

A recent improvement to *PE* approaches is the framework proposed by Khattab et al. (2023). This framework, called "Declarative Self-improving Language Programs, pythonically" (DSPy), learns how to combine different prompting and finetuning techniques to improve the generated answer. They show that their discrete optimization leads to better output compared to expert-designed prompts, even when applied to smaller LLMs. Their approach is text-based and revolves around generating examples of the desired output and then tuning the prompt and the LLM itself based on these examples. In spirit, their approach is similar to hyper-parameter tuning, in that it uses a type of (small) training data and a performance metric (e.g. whether the answer is an exact match to a certain label), and then adjusts the prompt and the LLM to maximize the scoring of its answer on this training set. The twist is, that the DSPy framework can perform such hyper-parameter tuning in an automized fashion, by generating candidate prompts itself and selecting from them. Our proposed approach differs from DSPy in that we optimize in a continuous space, and that our goal is to find a numeric summary for a document, rather than to generate a good response to language tasks, such as question answering.

GS: In this work, we propose *generative summaries* (GS) which optimize the input prompt to an LLM, such that we obtain a desired outcome. These *generative summaries* should perfectly replicate the document they summarize, thereby capturing all information contained in it. These document summaries are deterministic because we obtain them through maximum likelihood estimation and they live in continuous numeric space. We can evaluate the quality of such a document summary directly through the obtained likelihood.

Table 1: Overview of methods to summarize documents.

Method	Origin	Reference	Deterministic?	Generation?	Type?
BERT	Next-sentence prediction task	Devlin et al. (2018)	✓	×	Numeric
PWE	Aggregation of token information	Shen et al. (2018)	✓	×	Numeric
PE	Emergent capability of LLM	Khattab et al. (2023)	×	✓	Textual
GS	Maximize the likelihood to regenerate the focal document	***Proposed method***	✓	✓	Numeric

To validate whether our proposed approach is a viable alternative to these established methods, we will evaluate three aspects of our generative summaries on an artificial dataset, and explore their use in marketing applications. This yields these three *research objectives* for the validation and one explorative objective for the marketing application:

1) Can we use these generative summaries for classification tasks?
2) Can we regenerate the summarized document with our generative summaries?
3) Can we generate new documents based on our generative summaries?
4) *Explorative*: Can these generative summaries deliver useful insights in market research?

Methodology

The probability of generating the sequence $t_1, \ldots, t_T$ with the LLM when using the embedding $\boldsymbol{s}$ as an input is $p(t_1, \ldots, t_T \mid \boldsymbol{s})$. We can split this probability into conditional probabilities due to the autoregressive architecture of the LLM and because we observe the target sequence. This provides large computational gains, as we can calculate these parts in parallel. Hence, we define the log-likelihood of the generative summary for the target sequence as

$$\mathcal{L}(\boldsymbol{s} \mid t_1, \dots, t_T) = \log p(t_1|\boldsymbol{s}) + \log p(t_2|t_1, \boldsymbol{s}) + \dots + \log p(t_T|t_1, \dots, t_{T-1}, \boldsymbol{s}),$$

and find the optimal generative summary as

$$\mathbf{s}^* = \arg\max_{s} \mathcal{L}(\boldsymbol{s} \mid t_1, \dots, t_T). \tag{1}$$

We summarize the training process for these single generative summaries in Algorithm 1. The generative summary is a vector of length E, which is the same dimension as the LLM's input embedding.

Algorithm 1 Training of Single Summary Embeddings

```
i ← 0
ε ← 0.01
s ← Initialization(·)
while True do
    l^(i) ← L(s | t_1, ..., t_T)
    ∇_s^(i) l ← ComputeGradient(l^(i))
    s^(i+1) ← Optimizer(s^(i), ∇_s^(i) l)
    i ← i + 1
    if l^(i) < ε then
        break
    end if
end while
```

To regularize these embeddings and make them more interpretable, we are interested in finding a low-dimensional sub-space in which we can represent the D advertising claims. For this, we use a factor model and estimate the generative summaries for all advertising claims jointly, yielding the $D \times E$ matrix of generative summaries $\mathbf{S}$. We restrict the degrees of freedom to F hidden factors. To implement this factor model, we create an autoencoder with one hidden layer that is fully connected with linear activation functions (Goodfellow, Bengio, and Courville 2016). The input data to this autoencoder is an identity matrix of dimension D, making it a factor model. Before passing the generative summary to the LLM, we normalize them with layer normalization (Ba, Kiros, and Hinton 2016). Layer normalization works across the hidden nodes, rather than across the observations in a batch (batch-normalization, see Ioffe and Szegedy 2015). Thereby, it can also be used with a single observation. In the optimization, we encounter sharp spikes in our gradients, that originate in the decoder of the model. We mitigate these by gradient clipping (Zhang et al. 2020). We illustrate this neural network representation of the factor model in Figure 1.

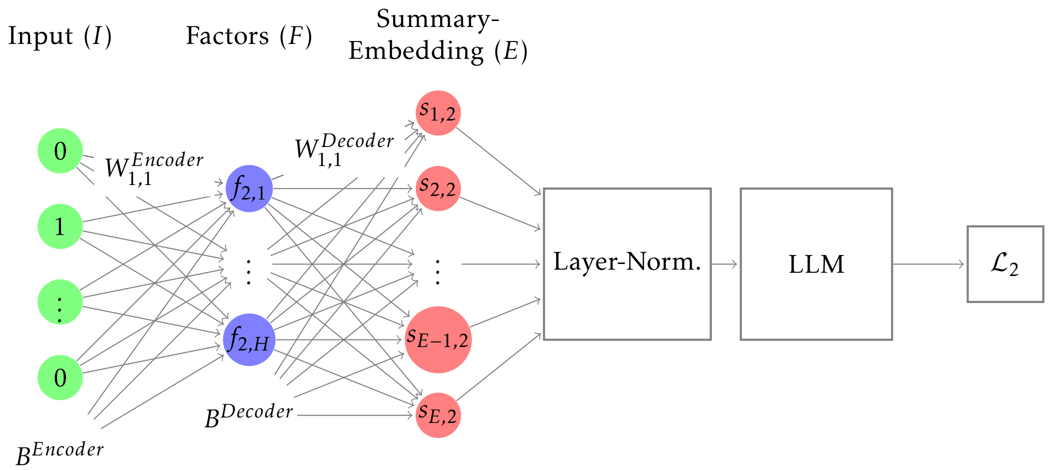


Figure 1: Factor model in Neural Network form, example for document 2

We denote the matrix of encoder and decoder weights by $\mathbf{W}^{Encoder}$ and $\mathbf{W}^{Decoder}$. These matrices have the dimensions $D \times F$ and $F \times E$. We also define the encoder and decoder biases as $\boldsymbol{b}^{Encoder}$ (length F) and $\boldsymbol{b}^{Decoder}$ (length E) and stack them into the matrices $\mathbf{B}^{Encoder}$ and $\mathbf{B}^{Decoder}$[9]. We denote the resulting factor model by $\text{Factor}_{\Omega_{Factor}}(\mathbb{I}_D) : \mathbb{I}_D \rightarrow \mathbb{R}^{D \times E}$ and collect the weights and biases in Ω_{Factor}.

When we define the weights and biases in this form, we can write the factor representation of the input as

$$\mathbf{F} = \mathbf{W}^{Encoder} + \mathbf{B}^{Encoder},$$

thereby the matrix of generative summaries for all claims becomes

$$\mathbf{S} = \left(\mathbf{W}^{Encoder} + \mathbf{B}^{Encoder}\right)\mathbf{W}^{Decoder} + \mathbf{B}^{Decoder}.$$

When we estimate the matrix of generative summaries $\mathbf{S}$, we use the joint log-likelihood to re-generate all documents in the dataset, $\mathcal{L}_D$. We estimate the generative summaries by maximizing this joint log-likelihood with respect to the parameters of the factor model:

$$\mathbf{S}^* = \underset{\mathbf{W}^{Encoder}, \mathbf{W}^{Decoder}, \mathbf{B}^{Encoder}, \mathbf{B}^{Decoder}}{\arg\max} \mathcal{L}_D(\mathbf{S}). \tag{2}$$

We summarize the training process for finding the sub-space representation in Algorithm 2.

[9] We do this by taking the Kronecker product with a length D vector of ones, $\mathbf{1}_D$, to obtain the biases in matrix form as $\mathbf{B}^{Encoder} = \mathbf{1}_D \otimes \boldsymbol{b}^{Encoder}$ and $\mathbf{B}^{Decoder} = \mathbf{1}_F \otimes \boldsymbol{b}^{Decoder}$.

Algorithm 2 Training of Summary Embeddings based on factor model

$i \leftarrow 0$
$\epsilon \leftarrow 0.01$
$\mathbf{W}_{(i)}^{Encoder}, \mathbf{W}_{(i)}^{Decoder}, \mathbf{B}_{(i)}^{Encoder}, \mathbf{B}_{(i)}^{Decoder} \leftarrow \text{Initialization}(\cdot)$
while $True$ **do**
  $\mathbf{S} \leftarrow \left(\mathbf{W}_{(i)}^{Encoder} + \mathbf{B}_{(i)}^{Encoder}\right)\mathbf{W}_{(i)}^{Decoder} + \mathbf{B}_{(i)}^{Decoder}$
  $l_{(i)} \leftarrow \mathcal{L}_D(\mathbf{S})$
  $\nabla_{(i)} \leftarrow \text{ComputeGradient}\left(l^{(i)}\right)$
  $\mathbf{W}_{(i+1)}^{Encoder}, \mathbf{W}_{(i+1)}^{Decoder}, \mathbf{B}_{(i+1)}^{Encoder}, \mathbf{B}_{(i+1)}^{Decoder} \leftarrow \text{Optimizer}\left(\mathbf{W}_{(i)}^{Encoder}, \mathbf{W}_{(i)}^{Decoder}, \mathbf{B}_{(i)}^{Encoder}, \mathbf{B}_{(i)}^{Decoder}, \nabla_{(i)}\right)$
  $i \leftarrow i + 1$
  if $l_{(i)} < \epsilon$ **then**
    break
  end if
end while

For this project, we obtained a SURF NWO Small Compute Grant[10] and perform all computations on the Snellius High-Performance Cluster[11]. We train these generative summaries until we obtain a likelihood to generate the target claim of 0.99. We use the Adam optimizer (Diederik P. Kingma et al. 2014) with a learning rate of 0.1, no weight decay, and the values of 0.9 and 0.999 for the exponential decay rates of the first and second moment respectively. To initialize our generative summary and factor model, we use the standard initialization of PyTorch[12]. We train on a Nvidia A100-GPU, which takes about 30 seconds for 20 claims. The training of the factor representation with 2 factors and for the same claims is more computationally intense and takes about 60 minutes on the same computer. Here we train for a joint likelihood of 0.99. We always verified that the generative summaries regenerate all focal claims correctly.

Diagnostics

As benchmarks for our model, we use the BERT and PWE methods. Each of these methods yields one embedding vector per document. For the PWE method, this embedding is of length 100 and for the BERT method, it is of length 768. We obtain the PWE embedding, by applying Word2Vec embedding to each token in the document and averaging out over the tokens (Mikolov, Ilya Sutskever, et al. 2013; Shen et al. 2018). For the BERT method, we pass the document through the pre-trained BERT model and extract the prediction for the CLS token, which serves as the document summary.

To analyze our results, we will use a suite of diagnostic tools, which we will explain here briefly.

First, we form two correlation matrices. One between the claims $\mathbf{C}^{Claims}$ and one between the embedding dimensions $\mathbf{C}^{Embeddings}$. For the former, the correlation at element i, j of the correlation matrix is the correlation of the i-th and j-th claim:

$$\mathbf{C}_{i,j}^{Claims} = \frac{\sum_{k=1}^{E}\left(\mathbf{S}_{i,k} - \frac{1}{E}\sum_{k=1}^{E}\mathbf{S}_{i,k}\right)\left(\mathbf{S}_{j,k} - \frac{1}{E}\sum_{k=1}^{E}\mathbf{S}_{j,k}\right)}{\sqrt{\sum_{k=1}^{E}\left(\mathbf{S}_{i,k} - \frac{1}{E}\sum_{k=1}^{E}\mathbf{S}_{i,k}\right)^2 \sum_{k=1}^{E}\left(\mathbf{S}_{j,k} - \frac{1}{E}\sum_{k=1}^{E}\mathbf{S}_{j,k}\right)^2}}.$$

For the latter, the correlation at element i, j of the correlation matrix is the correlation of the i-th and j-th embedding dimension:

$$\mathbf{C}_{i,j}^{Embeddings} = \frac{\sum_{k=1}^{D}\left(\mathbf{S}_{k,i} - \frac{1}{D}\sum_{k=1}^{D}\mathbf{S}_{k,i}\right)\left(\mathbf{S}_{k,j} - \frac{1}{D}\sum_{k=1}^{D}\mathbf{S}_{k,j}\right)}{\sqrt{\sum_{k=1}^{D}\left(\mathbf{S}_{k,i} - \frac{1}{D}\sum_{k=1}^{D}\mathbf{S}_{k,i}\right)^2 \sum_{k=1}^{D}\left(\mathbf{S}_{k,j} - \frac{1}{D}\sum_{k=1}^{D}\mathbf{S}_{k,j}\right)^2}}.$$

[10] https://www.surf.nl/en/small-compute-applications-nwo
[11] https://www.surf.nl/en/services/snellius-the-national-supercomputer
[12] https://github.com/pytorch/pytorch

Besides these correlation matrices, we also use Principal Component Analysis (PCA). For a data matrix $\mathbf{X}$, PCA extracts k principal components by taking the eigenvectors that correspond to the k largest eigenvalues of the matrix $\mathbf{X}^T\mathbf{X}$. These k eigenvectors contain the coordinates of the data points in the k-dimensional space spanned by these principal components (Pearson 1901).

Data

As a first evaluation, we create a small synthetic dataset of 20 advertising claims for hair shampoo by querying ChatGPT[13] with the prompt in Figure 2. We want half of these claims to advertise tangible aspects of the product and the other half to advertise intangible aspects of the product. These advertising claims all focus on how "shiny" the product ones hair makes. As as robustness check, we repeated this task for different attributes (healthiness and colorfulness), and a different product category (surface cleaners). In total, we obtain 80 advertising claims, 20 for each of these settings. We show the shiny hair shampoo claims in Table 2 and moved the other analyses to the appendix.

> **Prompt:** *I work for a marketing agency. I want to market a haircare product and need some claims for this.* ***I want you to emphasize how shiny the haircare product consumers' hair makes****. I need claims that differ in style with respect to how tangible the claims are. Make these claims brief. Can you give me 10 claims that are rather tangible and 10 claims that are rather intangible?*

Figure 2: Prompt to generate the benchmark advertising claims[14]

Table 2: Tangible (T) and Intangible (I) advertising claims for hair shampoo products. (Created by ChatGPT)

Number	Claim
0 (T)	Experience 50% more visible shine after just one use.
1 (T)	Formulated with light-reflecting technology for a glossy finish.
2 (T)	Transform dull strands into radiant, luminous locks.
3 (T)	Infused with nourishing oils that enhance natural shine.
4 (T)	See instant brilliance with our advanced shine-boosting formula.
5 (T)	Locks in moisture to amplify hair's natural luster.
6 (T)	Achieve salon-quality shine without leaving home.
7 (T)	Visible reduction in dullness, replaced with stunning shine.
8 (T)	Say goodbye to lackluster hair, hello to mirror-like shine.
9 (T)	Clinically proven to enhance shine by up to 70%.
10 (I)	Elevate your confidence with hair that gleams under any light.
11 (I)	Embrace the allure of luminous hair that turns heads.
12 (I)	Unleash the power of radiant hair that speaks volumes.
13 (I)	Transform your look with hair that exudes brilliance.
14 (I)	Feel the difference of hair that shines with vitality and health.
15 (I)	Rediscover the joy of hair that beams with inner vibrancy.
16 (I)	Indulge in the luxury of hair that shimmers with elegance.
17 (I)	Step into the spotlight with hair that radiates beauty.
18 (I)	Experience the magic of hair that dazzles with every movement.
19 (I)	Unlock the secret to hair that shines from within, reflecting your inner glow.

[13] https://chat.openai.com/

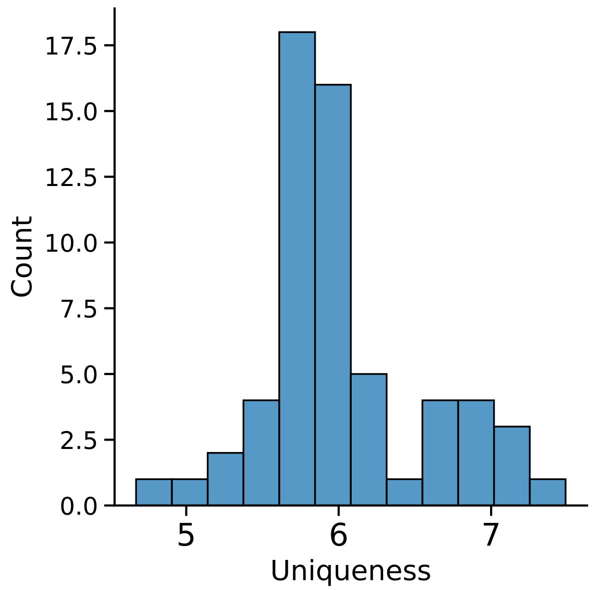
For our empirical application, we obtain two datasets from a market research company. The first one contains advertising claims for yogurt drinks and their evaluation by consumers. The second dataset contains advertising claims for yogurts and the design motivations behind these advertising claims, such as whether a claim talks about the natural ingredients of the yogurt or its local origin. These data stem from different variations of choice-based conjoint studies (Eggers et al. 2022), which we cannot further disclose due to confidentiality agreements. The data includes choice experiments aggregated at the level of the advertising claims, so we do not observe individual responses. We focus on measures of a claim's uniqueness and appeal, as perceived by the consumers. The advertising claims were designed for markets in different countries and are (translated into) English. Each dataset contains only one brand.

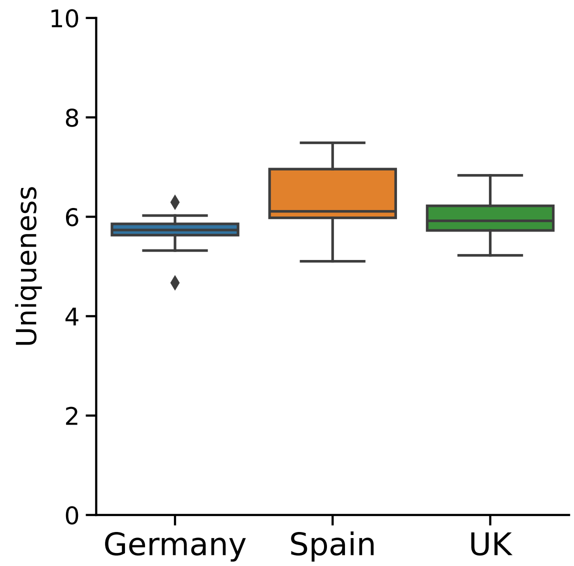
For the yogurt drinks, we have 60 aggregate claim evaluations, 20 for each country (Germany, UK, Spain). Many of the researched advertising claims occur in multiple of these separate studies. In total, we have 26 unique advertising claims for yogurt drinks. Most of these advertising claims get a uniqueness measure of around 5.8, with the whole distribution ranging from 4.8 to 7.4 on 10 point scale. When we look at the uniqueness measure concerning the country in which the study was conducted, we see that the variance is the smallest for the German market and the largest for the Spanish market. However, the means are not statistically different across the countries. The distribution of the rating measure is similar in shape, ranging from 43 to 80 on a 100 point scale. The rating and uniqueness are positively correlated (correlation of 0.8843). The yogurt drink advertising claims are similar to these synthetic claims:

- *"A burst of fresh flavor to energize your morning"*
- *"Refresh your day with a lively new taste"*
- *"Dynamic flavor for an invigorating start"*
- *"Begin your day with a crisp and revitalizing taste"*
- *"Revitalize your senses with a pure, fresh flavor"*

For the second set of claims, the claims for yogurts, we have 215 observations of 82 unique advertising claims. The data stems from studies in the UK, France, Germany, Spain, and Sweden. This data also contains a design motivation for the advertising claims. In total, there are 44 different classes for these design themes. We focus on the five largest classes: "Packaging", "Local & Responsible", "Sourcing", "Sustainability", and "Naturality", which account for 70% of the observations. Below, we show a short description of what these themes look like:

- **Packaging**: The claim emphasizes e.g. that the packaging is recyclable or made from eco-friendly materials
- **Local & Responsible**: The claim talks about the local origin of the yogurt or how the yogurt company supports local farmers
- **Sourcing**: The claim emphasizes the origin of the ingredients
- **Sustainability**: These claims state that the yogurt is e.g. climate-friendly
- **Naturality**: These claims state that the yogurt is e.g. made from natural ingredients or has been processed as little as possible

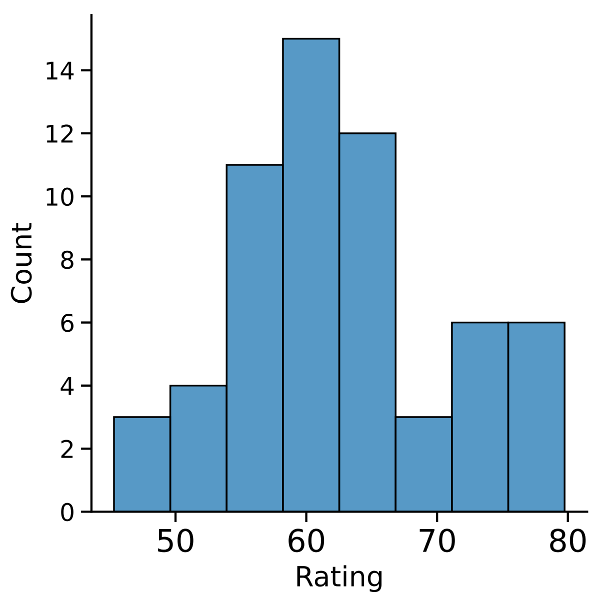


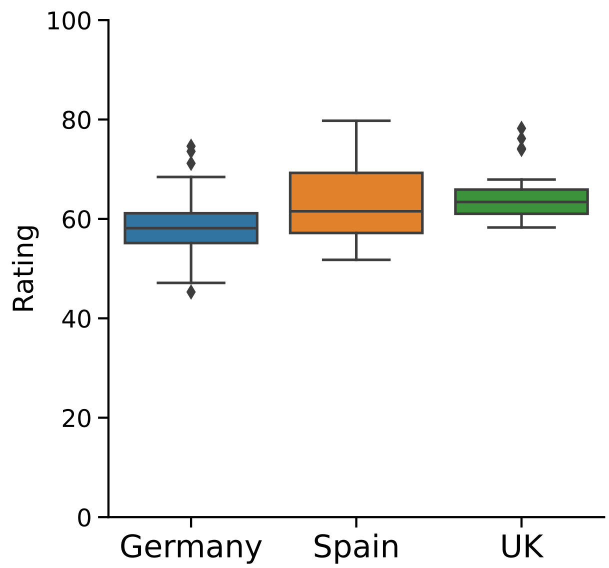


(a) Distribution of the uniqueness measure across all countries

(b) Distribution of the uniqueness measure per country. The black line indicates the mean.

Figure 3: Distributions for the claim uniqueness measure across the whole sample and as a boxplot per country.





(a) Distribution of the uniqueness measure across all countries

(b) Distribution of the appeal rating per country. The black line indicates the mean.

Figure 4: Distributions for the appeal rating across the whole sample and as a boxplot per country.

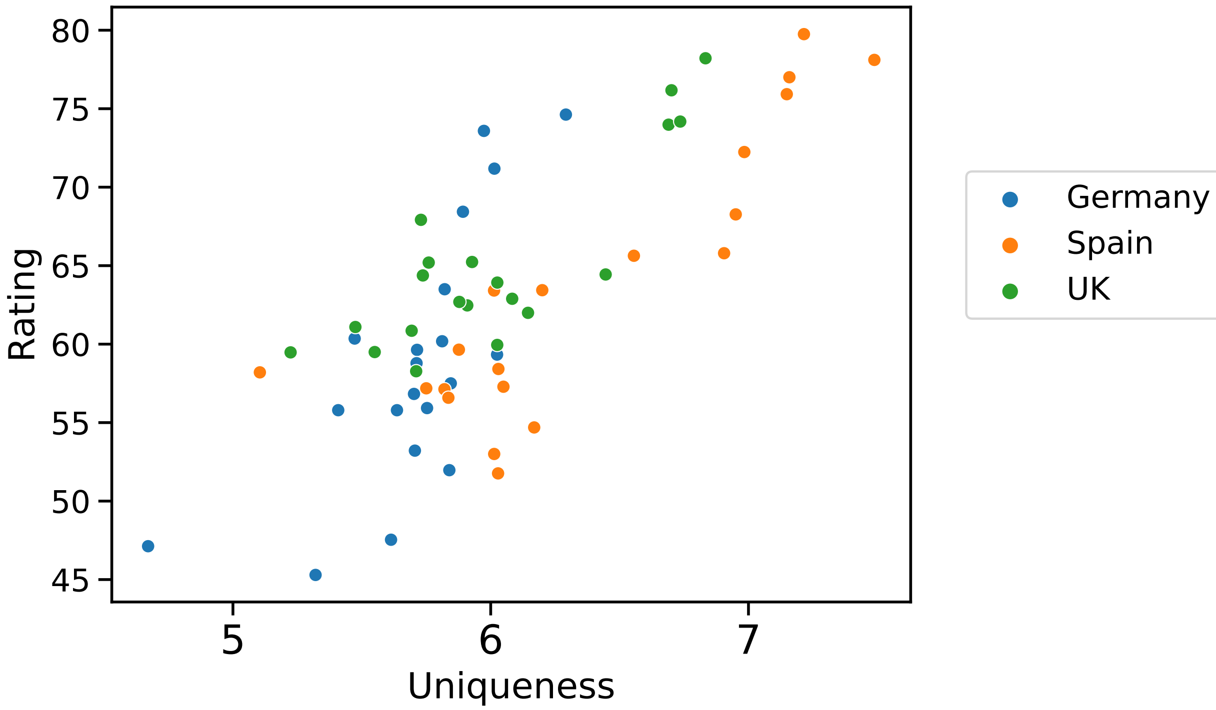


Figure 5: Scatterplot of uniqueness and rating , colored by country.

Results

We start by applying our method to the synthetic dataset of hair shampoo claims, that differ by whether they are formulated in a tangible or intangible way. We estimate these generative summaries in 3 versions: The single estimations, which estimate the generative summaries for each claim separately and without the factor structure (compare Equation 1), and two-factor models with two factors each (compare Equation 2). One of these uses the Layer Normalization, while the other one does not. We present these three versions, to illustrate the effects of the factor structure and the layer normalization. In the appendix, we show the same diagnostics for other datasets, such as advertising claims for hair shampoo that emphasize how healthy/colorful the hair becomes (instead of shinyness), as well as for a surface cleaner product. These results are robust to changing the order of the claims in the dataset. Figure 6 shows the training process for the generative summaries estimated with the 2-factor model with layer normalization. The y-axis shows the negative log-likelihood (a value of 0.01 corresponds to a likelihood of $e^{-0.01} = 0.99$). Despite our efforts to mitigate spikes, we still see some smaller spikes in the tail of the optimization.

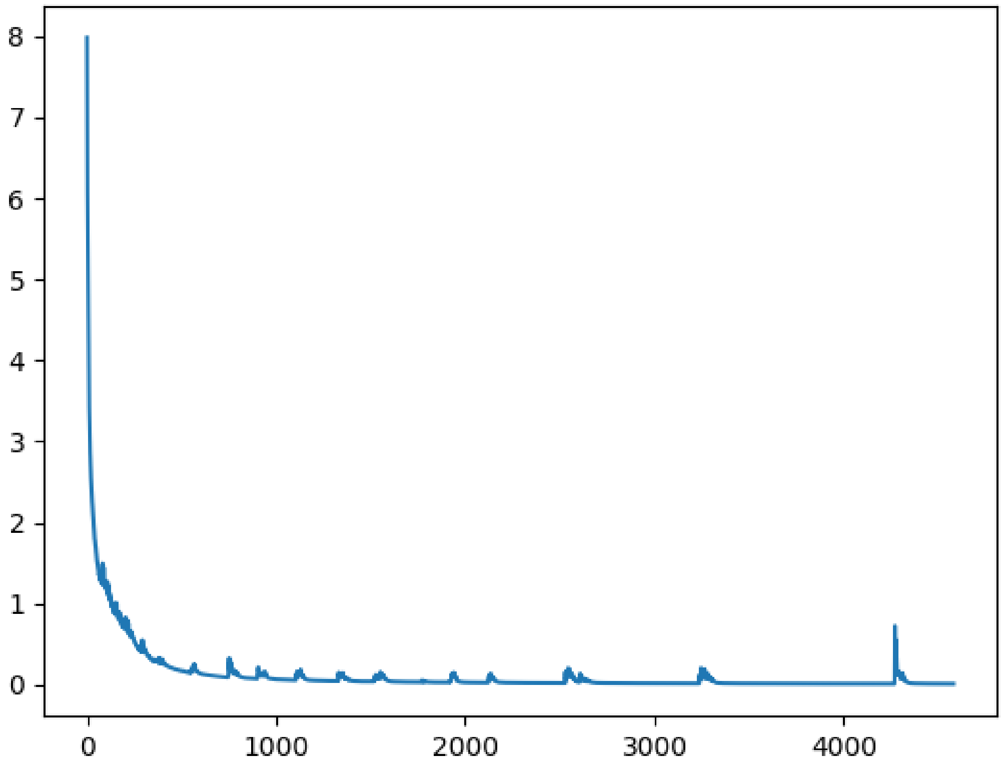
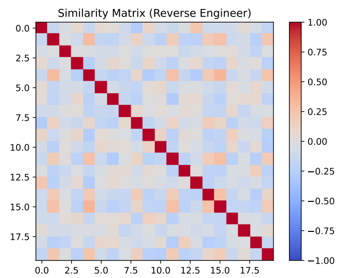


Figure 6: Training of embeddings

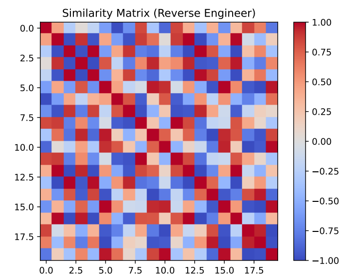
Objective 1: Can we use these generative summaries for classification tasks?

If our generative summaries capture relevant information in the advertising claims, then we can separate between the tangible and intangible advertising claims. To be left with only the difference due to being tangible or intangible we demean the embeddings across the embedding dimensions (compare Mikolov, Ilya Sutskever, et al. 2013). We then calculate the correlation matrix across the claims and the embedding dimensions. When visualizing these correlations across the claims, we expect the upper-left square and the lower-right square of the matrix to contain positive correlations, since these are the correlations for the claims of the same class (i.e. tangible-tangible or intangible-intangible). On the contrary, the upper-right and the lower-left quartile should contain negative correlations, since these are correlations between the two classes (tangible-intangible).

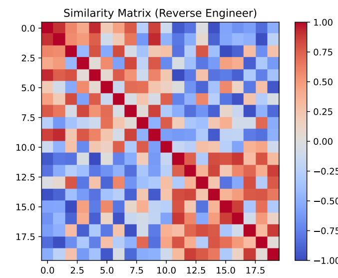
In Figure 7 we present these correlation matrices for our GS method and the PWE and BERT approaches. These correlation matrices show how the embedding values of two claims vary with each other. Identifying these two classes works well for the generative summaries which we estimate with 2 factors and layer normalization. When we compare this version to the PWE and BERT approaches, we see this pattern for all methods, indicating that all four methods can capture this part of the information in the advertising claim. We observe, that the correlations are the largest (in absolute value) for the generative summaries and only a few correlations are close to zero. In the appendix, we show the claim correlation matrices for our other generated datasets. The results mirror the results for the hair shampoo claims that emphasize the shinyness of the hair, with one exception: The hair shampoo claims are about how colorful it makes the hair. Here, we find that 2 factors are not enough to separate between tangible and intangible claims. We can recover these two classes by using a 15-factor model instead. Here we also observe, that the correlations are then weaker, as these document embeddings have more degrees of freedom in how their values come to be.



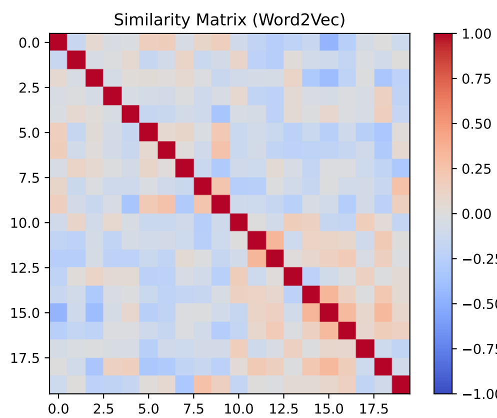
(a) GS: Single estimation



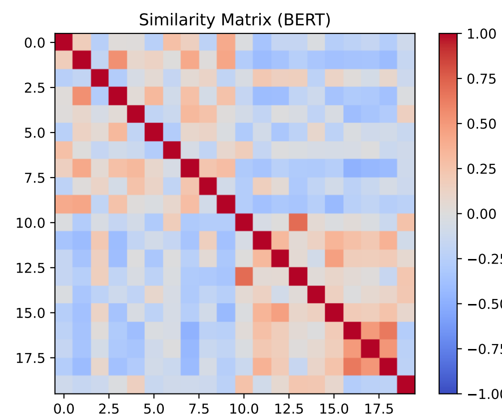
(b) GS: 2 Factors



(c) GS: 2 Factors with LayerNorm.



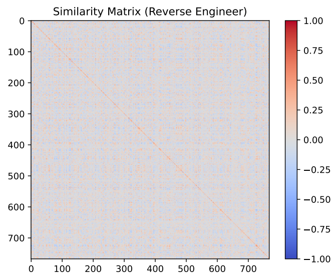
(d) PWE



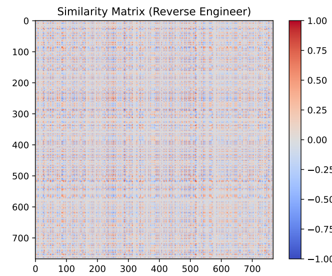
(e) BERT

Figure 7: Correlation matrices along the claims.

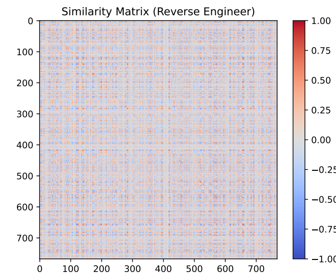
Figure 8 shows correlation matrices across the embedding dimensions and illustrates the effects of using a factor structure on the generative summaries. Each cell shows the correlation between two embedding dimensions calculated across the 20 advertising claims. When comparing our generative summaries with the two benchmarks, we can see a stronger grid pattern with more correlations that are removed from zero. This stems from our factor structure, as this pattern is not present in the single estimation version. Making many of the embedding dimensions linearly dependent strengthens the correlations between them. In contrast, for the BERT embedding most dimensions are barely correlated with each other. The Word2Vec embedding, which is used in PWE, only has 100 dimensions, as opposed to the 768 dimensions of our generative generative summaries and the BERT embedding. Its correlations appear to be stronger but do not exhibit the same grid pattern as for the generative summaries.



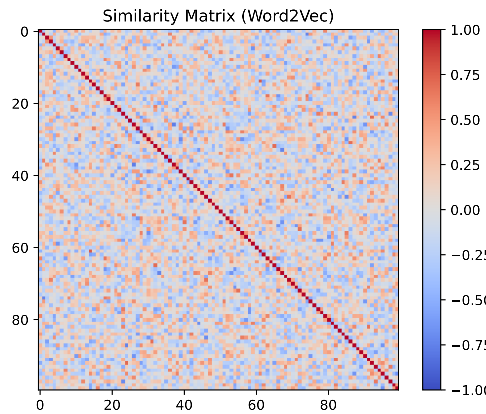
(a) GS: Single estimation



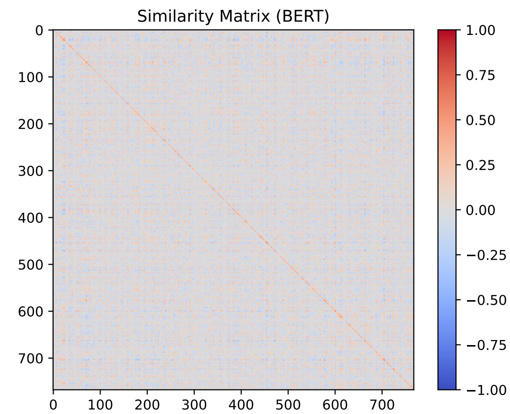
(b) GS: 2 Factors



(c) GS: 2 Factors with LayerNorm.



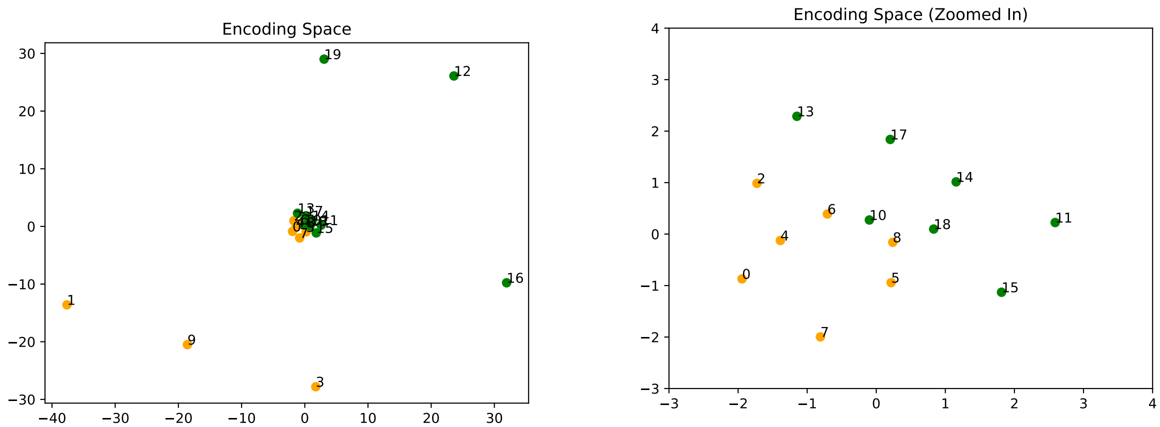
(d) PWE



(e) BERT

Figure 8: Correlation matrices along the embedding dimensions.

After establishing that our generative summaries capture similar information to established document summary techniques, we want to explore the low-dimensional factor space of our model. Figure 9 shows the low dimensional sub-space in which we located the advertising claims, which we color in orange (tangible claims) and green (intangible claims). The assigned numbers correspond to the number of the claims in Table 2. It appears that we can perfectly separate the two classes with the two hidden factors. Many of the points are close to zero, with a few of the claims being far removed from the other points, e.g. claims 1, 3, and 9 for the tangible, and claims 12, 16, and 19 for the intangible claims. When comparing distances in the encoding space, we can find some patterns that relate to the types of words that are used in the advertising claims. For example, the triangle of claims 0, 4 and 7 uses words related to vision ("... visible ...", "See ...", and "Visible ..."), and the two claims 2 and 13 both start with the word "Transform." The tangible claims far removed from the center, claims 1, 3, and 9, have in common that they talk about a form of "technology", whereas claims 3 and 9 both use the phrase "enhance [...] shine". For the intangible claims, the two claims 12 and 19 both start with the close words "Unleash" and "Unlock". However, there are also structures in this representation which are not intuitive. We would anticipate that claim 4 would have been close to these 3 tangible claims in the south-west of the plot, as it is about an "advanced shine-boosting formula", or the outlier claim 16 ("Indulge in the luxury of hair that shimmers with elegance"), would be part of the cluster at the center because it is similarly constructed as the 7 intangible claims there (pattern: *Feeling/Experience of the consumer*—"of hair that"—*feature of the hair*).



(a) All claims estimated by the factor model

(b) Zoom-in on the center claims estimated by the factor model

Figure 9: Two-dimensional factor space of the advertising claims. Numbers correspond to the numbers in Table 2, orange points are tangible claims, and green points are intangible claims.

Objective 2: Can we use these generative summaries to re-generate the focal document?

While our generative summaries appear to capture the information contained in a document, one of their main advantages is that we can use them for generation. In the following, we analyze the generation with the generative summary. If our generative summaries indeed maximize the likelihood of generating the target text, then the most likely token at each position in the generated sequence should be the corresponding token from the summarized document. We expect that at each generation step, the probability for the correct token approaches one, while the probability for all other tokens goes to zero. In Figure 10, we visualize the next token probabilities at each generation step for the correct token. We see that the probability for the correct token is close to one, with the first token in a sequence having the lowest probability of being generated correctly. This holds across all claims. An explanation for this is, that at the first position, the generation relies on the generative summary alone. At later stages the joint information of generative summary and previously generated tokens stabilizes the generation, driving the generation probabilities closer to 1.

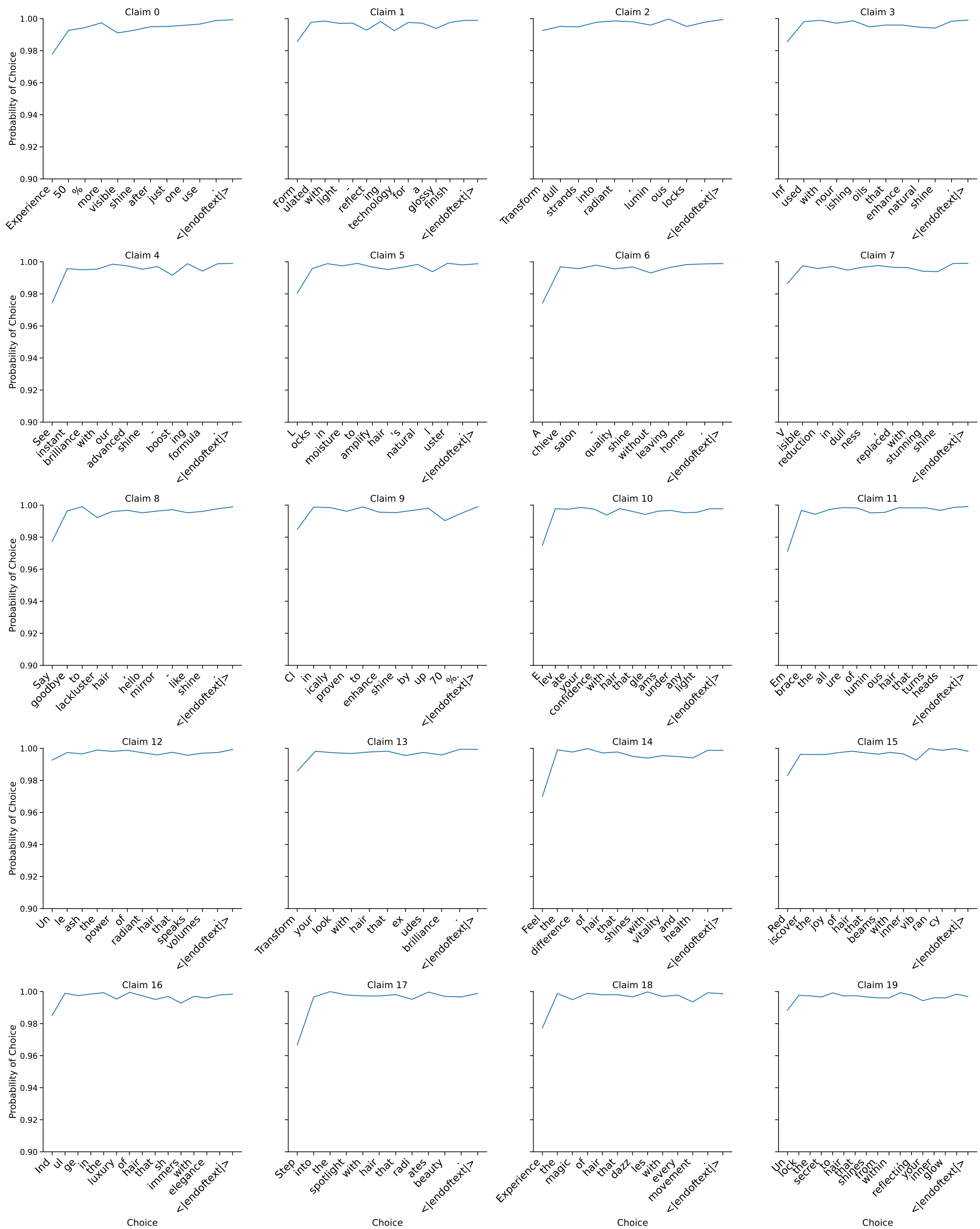


Figure 10: Conditional generation probability for the most likely token at each step for the factor model. All tokens match the target claims.

Next, we explore also the generation probability for the 2nd and 3rd most likely token at each position, to get a better understanding of the distribution at each step. We present two examples. One for the generation of the claim with the lowest probability to correctly generate the first token, in Table 3, and another example that illustrates how we benefit from the LLMs pre-training, in Table 4. Both tables show these generation probabilities for the three most likely tokens at each generation step. Table 3 again shows how the critical step in the generation is the first token of the sequence, as the probability to generate the correct token in the second position jumps for 0.8876 to 0.9925 and never drops lower than 0.9875 again. In Table 4, we see that the most likely alternatives to the first token, "Experience", are "Form" and "See". We find similar examples for the second token (1st Choice: "50", 2nd Choice: "25", 3rd Choice: "20") and the fifth position (1st Choice: "visible", 2nd Choice: "protective", 3rd Choice: "shine"). The fact that these tokens get the second and third-highest generation probabilities is likely due to the pre-training of the LLM, and reflects how these words tend to be used in similar situations, such as in shampoo product claims.

Table 3: Example for a low 1st probability based on factor model

	3rd	2nd	Choice	Prob. 3rd	Prob. 2nd	Prob. Choice
0	Feel	Un	Step	0.0287145	0.0344409	0.887634
1	onto	back	into	0.000927151	0.00125316	0.992544
2	to	class	the	5.98035e-05	6.91189e-05	0.99934
3	'	sun	spotlight	0.000623843	0.00123234	0.992407
4	into	in	with	0.00192485	0.00199383	0.992454
5	a	hairs	hair	0.000946804	0.000969693	0.995565
6	above	.	that	0.000743194	0.0014022	0.993072
7	X	can	radi	0.000538268	0.00169703	0.992411
8	ating	ats	ates	0.000915091	0.00206985	0.996706
9	heat	into	beauty	0.00111988	0.00228065	0.987492
10	).	.)	.	0.000529124	0.00114012	0.996219
11		]	*eos-token*	0.000603324	0.00100365	0.995909
12	*linebreak*	*linebreak*	*eos-token*	7.61097e-06	0.00124311	0.99872
13	*linebreak*	*linebreak*	*eos-token*	2.16886e-05	0.000565556	0.999231
14	*linebreak*	*linebreak*	*eos-token*	3.80865e-05	0.00084979	0.998713
15	*linebreak*	*linebreak*	*eos-token*	5.16162e-05	0.00161664	0.997724
16	*linebreak*	*linebreak*	*eos-token*	0.000101122	0.0031923	0.995816

Table 4: Example for similar tokens based on factor model

	3rd	2nd	Choice	Prob. 3rd	Prob. 2nd	Prob. Choice
0	See	Form	Experience	0.00921188	0.013993	0.943786
1	20	25	50	0.00203264	0.00445767	0.976235
2	%.	.	%	0.00142157	0.00205115	0.994173
3	better	More	more	0.000946993	0.00160339	0.996037
4	shine	protective	visible	0.00178126	0.00218073	0.979003
5	glow	light	shine	0.000826363	0.00209097	0.983356
6	when	by	after	0.00171736	0.00367565	0.987
7	a	well	just	0.000570301	0.00172336	0.98959
8	two	a	one	0.00153683	0.00186334	0.989587
9	shine	usage	use	0.000427258	0.00324287	0.990624
10	,	!	.	0.000354906	0.000630647	0.9981
11	.		*eos-token*	0.000454232	0.00156762	0.994366
12	Transfer		*eos-token*	0.000179846	0.00032993	0.998234
13	*linebreak*	More	*eos-token*	0.000109437	0.000194042	0.998748
14	More	*linebreak*	*eos-token*	7.64376e-05	0.000174342	0.999236

	3rd	2nd	Choice	Prob. 3rd	Prob. 2nd	Prob. Choice
15	Lab	*linebreak*	*eos-token*	5.29115e-05	0.000444251	0.998936
16	Lab	*linebreak*	*eos-token*	0.000276325	0.00155492	0.997141

Objective 3: Can we use these generative summaries to generate new documents?

To evaluate whether we can use our generative summaries to generate new documents, we want to explore the space in between two generative summaries. Below, we explore combinations of generative summaries and points from the factor space to see what kinds of documents they generate. We start by interpolating between two generative summaries which we have estimated by themselves, i.e. without a factor model connecting them. The second analysis will be on the factor space of the hair shampoo 2-factor model.

Table 5 shows the documents that we generate for a convex combination of the single generative summaries for the claims "*Unlock the secret to hair that shines from within, reflecting your inner glow.*" (A) and "*Rediscover the joy of hair that beams with inner vibrancy.*" (B), based on their generative summaries, which we estimated directly. We combine them as $\boldsymbol{s}_{combine} = weight * \boldsymbol{s}_A + (1 - weight) * \boldsymbol{s}_B$, where $weight \in [0, 1]$. For weights that are close to either 0 or 1 we still generate one of the two claims. When we step further into the middle between these two generative summaries the end of the sequence changes first (e.g. "*Unlock the secret to blackened nails that shine from the shine.*" at for $weight = \frac{3}{20}$). For weights that are close to $\frac{1}{2}$, we stop generating eos-tokens and the generated strings become unintelligible. However, these strings still maintain some of the structure of the focal claims such as having words related to beauty products and starting with the syllable "Un" or "Red".

Table 5: Interpolation between two advertising claims based on non-factor model.

	Weight	Generated String
0	0/20	Unlock the secret to hair that shines from within, reflecting your inner glow. *eos-token*
1	1/20	Unlock the secret to hair that shining from within, your bedroom. *eos-token*
2	2/20	Unlock the secret to black metal's glow. *multiple eos-token*
3	3/20	Unlock the secret to blackened nails that shine from the shine. *eos-token*
4	4/20	Un to of for the team members upon Moderation of the, that
5	5/20	Un to (93) of 15 + 18 + 36 Add to
6	6/20	Un to (mit. of private eye and nirvana-ly-ly
7	7/20	Uncle-healedered with a lifetime-changing blend of light and energy and
8	8/20	Unclelyed by the Tone's Bright Side™ Lipstick. Bright
9	9/20	Unclelying our clients and fory, we're sure about your smile.
10	10/20	Redmates on the Road to of Dreams. Inspirating your soul with
11	11/20	Redmates at the Sunlight Your Hair. *multiple eos-token*
12	12/20	Redhs founder, that a master the way she loves......
13	13/20	Redhs founder, of where, at, and to and to, and can be
14	14/20	Rediscover the joy of purpose and detail with a strand of your hair. (and
15	15/20	Rediscover the joy of success with long,istry's health and creativity. back to
16	16/20	Rediscover the joy of hair that loves to making you.™ the way.br
17	17/20	Rediscover the joy of hair that's at the-corrects life with a healthy
18	18/20	Rediscover the joy of hair that beams with inner vibrancy. *eos-token*
19	19/20	Rediscover the joy of hair that beams with inner vibrancy. *eos-token*

Next, we want to explore the encoding space of a factor model and generate new texts from points in this encoding space. In Figure 11, we perform a grid search across the factor space of our 2-factor model for shiny hair shampoo claims. For each point in the grid search, we generate a text by passing this point through the decoder to create a generative summary. We then pass this generative summary into the LLM to generate a text for this point from the encoding space. In the plot, we have three types of different markers. A blue bubble with a number represents the location of the respective claim, while a colored dot represents a point where we generate one of the points that are part of the training data. Red crosses are points from which we generate a string, that comes to an end with the end of text token and is not part of the training data ("candidate points"). Whitespace represents points where we do not generate a string that comes to an end within the number of tokens that we consider. Around each advertising claim, we find an island of points from which we can generate the same claim. The islands are surrounded by candidate points, and areas that are further removed from any of the training claims tend to be whitespace.

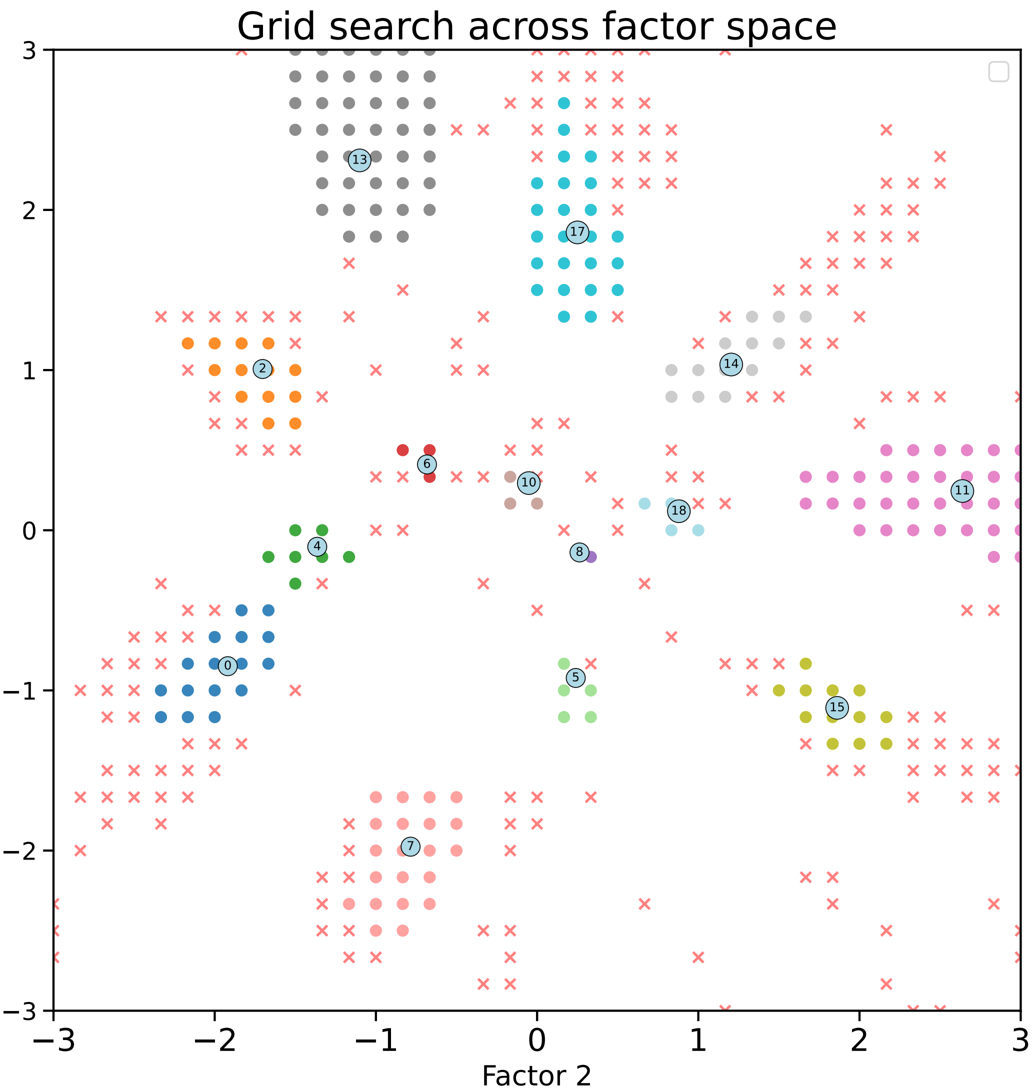


Figure 11: Exploration of Encoding Space of the factor model. Numbered bubbles are the locations of the respective claim, red crosses represent candidate points. These are points from which we generate a text that ends with the eos-token, but is not part of the training data. Points that do not generate an eos-token are marked by whitespace.

Some examples of candidate points are:

- “Experience 50% more visible shine after-changing lighting.<|endoftext|>”
- “Visible reduction in dullness, but with a touch of menace.<|endoftext|>”
- “Step into the spotlight with hair that to website in.).<|endoftext|>”
- “Elevate your confidence with hair that luxeaks.<|endoftext|>”
- “Experience the magic of-life.<|endoftext|>”

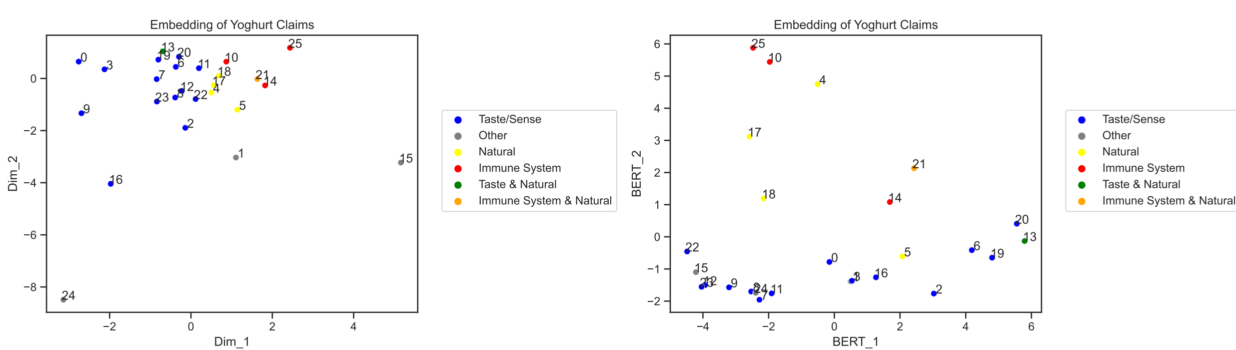
Many of these candidate points generate the same texts (if the points are close to each other). Most of these generated claims do not seem to be intelligible. However, they still tend to use words from the training data and have a sentence structure that is similar to product claims.

Objective 4: Empirical Application

After validating the properties of our method on a synthetic dataset, we now apply it to market research data, to illustrate how it can be used in practice. In the following, we look at two empirical applications of our generative summaries. The first application is on product claims for yogurt drinks and their appeal to consumers from three different countries. In the second application, we analyze product claims for yogurt. For this dataset, we also have design motivations behind these product claims, which we will use to validate whether our algorithm captures these design motivations. Since we cannot disclose the actual advertising claims, we generate a set of similar claims with ChatGPT[15] or use "stand-ins" for advertising claims, which are supposed to represent these and their properties without actually disclosing them. However, we performed all computations on the actual data.

Yoghurt-drink claims, word features, and consumer ratings (Application 1)

Besides the advertising claims themselves, we also have measures on the uniqueness of a claim (as perceived by consumers) and an overarching appeal rating of the claim by the consumer. To aid the market research process of finding the best advertising claims, we investigate the encoding space of the advertising claims, as estimated with a 2-factor generative summary. First, we explore this space, to identify linguistic features of the advertising claims. These are not features that were reported in the data but rather features that we interpret ourselves by looking at claims that are close to each other in the encoding space. We look at both word features (i.e. a certain word occurs in the claim) and a higher level theme of the claim, e.g. whether the claims frame the yogurt drink as a "breakfast" or "start in the day" ("morning" theme). Next, we are interested in whether the uniqueness measure is positively correlated to the distance of claims to each other in the encoding space. Namely, whether more unique claims are more separated from other claims in the encoding space. Finally, we explore whether certain regions of the encoding space are associated with higher appeal ratings. We trained this 2-factor document summary until we reached a joint likelihood of 0.99, which training took around 8 minutes on the Nvidia A100 GPU. We also embed these advertising claims by summarizing them with BERT and extract the first two principal components from the resulting embedding matrix (see Pearson 1901). We want to caution, that these findings are based on a very small sample and are supposed to serve as a proof of concept on how these generative summaries, or here their encoding space, can be useful in market research.

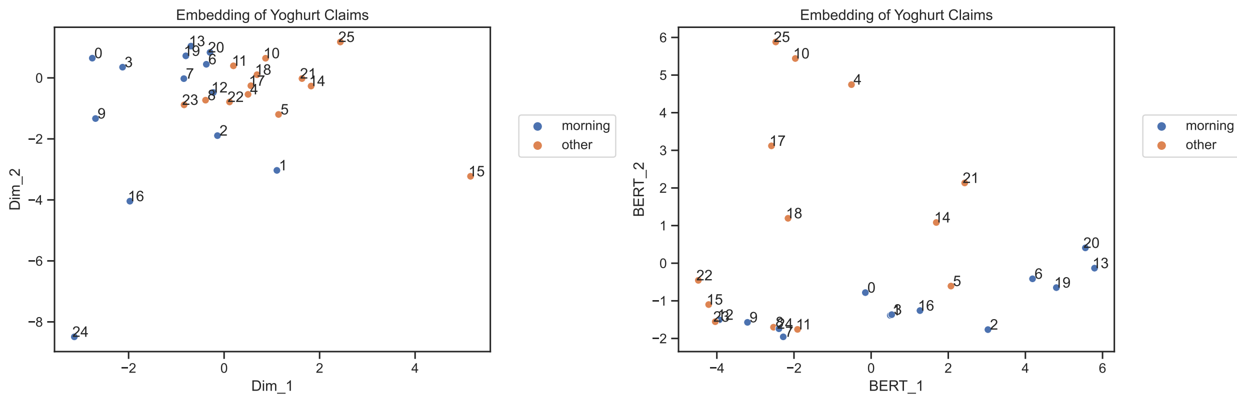


(a) Generative Summaries (factor model)

(b) BERT CLS + PCA

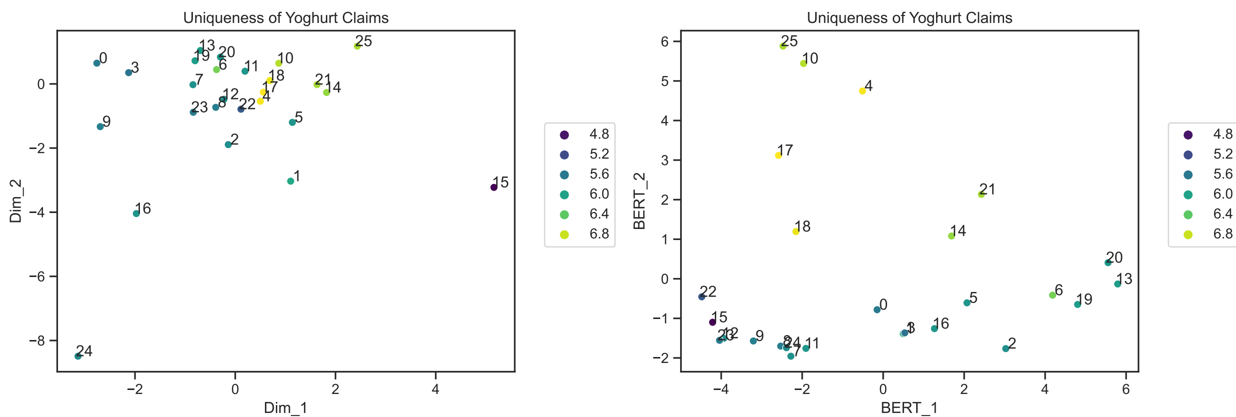
Figure 12: Embedding of the yogurt claims, colored by word features.

[15] https://chatgpt.com/



(a) Generative Summaries (factor model) (b) BERT CLS + PCA

Figure 13: Occurrence of words relating to the morning theme in the low-dimensional spaces.



(a) Generative Summaries (factor model) (b) BERT CLS + PCA

Figure 14: Uniqueness score of embedded yogurt drink claims.

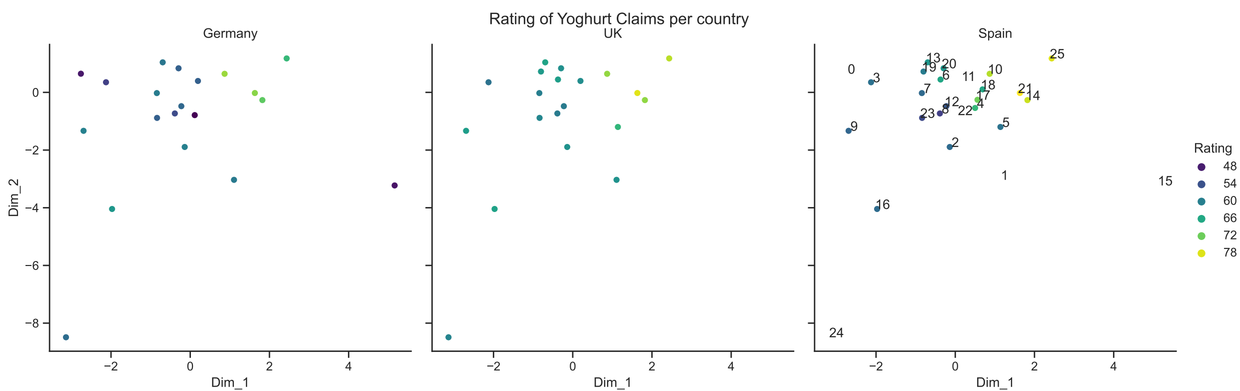
Figure 12 shows the encoding space for the yogurt drink advertising claims and illustrates how the embedding picks up on language features of the documents. Namely, Figure 12 colors the claims based on whether they contain the words "taste" or "sense", "natural", and/or "immune system". We color-coded these three by the basic colors (yellow, red, blue), and claims that contain multiple of these words by the mixtures of these colors. If a claim contains none of these words, we code it in grey. These "flag-words" come from looking at the factor space and comparing close advertising claims with each other, through this exploration we discovered these flag-words. Hence, this is not an external label.

Claims with the same word features live in similar regions of the encoding space, while claims that contain none of these words are pushed to the outside of the plot. Intuitively, we would expect that claims that contain multiple of these word features form a transition between the claims that only have one of the word features. For this, we only have claims 13 and 21 as examples, whereas the former does not form such a boundary, and the latter is at the edge of the "Natural" and "Immune System" classes. Despite not being an external label, these word features tend to cluster to certain regions in the BERT visualization as well.

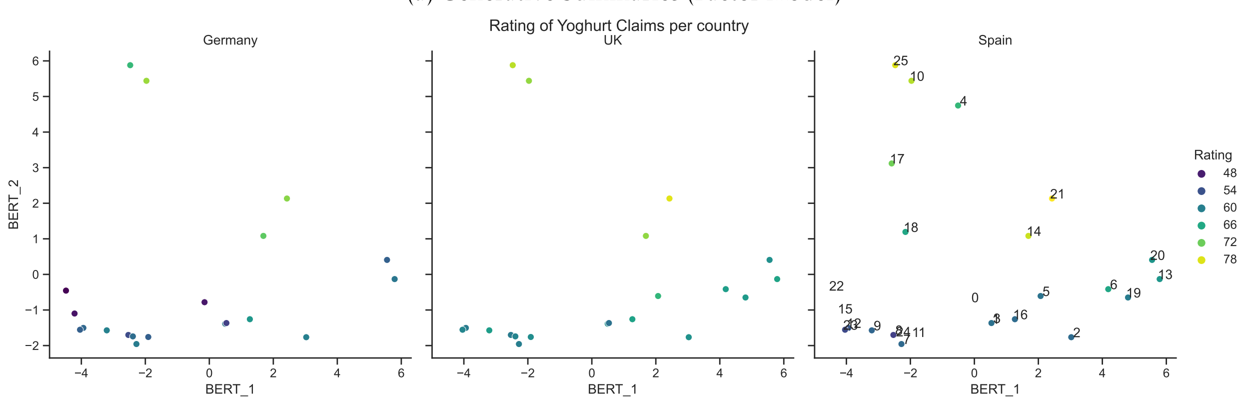
Similarly, Figure 13 colors the advertising claims blue if they are written with a "morning" theme. An example of such a claim would be "*Begin your day with a crisp and revitalizing taste*" (synthetic example). Like the word features in Figure 12, we hand-coded these themes. The split of the two groups is not crisp, but morning-themed advertising claims tend to occur in the south-west of the space, while the other claims are in the north-east. Similarly, for the BERT visualization, morning-themed claims occur mostly in the southeast of the plot.

We expect advertising claims that are more unique to have a more isolated position in the encoding space, as they are less similar to other claims. In other words, the uniqueness score should be positively correlated with the Euclidean distance of an encoded claim to its nearest neighbor. This is the case for the BERT-based space, however, not for our factor space. A potential reason for this could be that two factors are not enough to pick up on higher levels of abstraction, such as uniqueness, while the principal components of a 768-dimensional embedding could.

For the 2-factor generative summary space, Figure 14 shows that the uniqueness measure and nearest neighbor distance are negatively correlated ($r = -0.44$). Here the three most unique claims (4, 17, 18) cluster together, and the further claims are away from these three, the less unique they tend to be. Claims with low uniqueness scores are spread further apart and far away from the most unique claims. These three most unique claims have some things in common, all of these use a combination of the words “natural”, “active”, and talk about ingredients (with synonyms). On the other hand, claim 15 is the only claim in the dataset using the word “yogurt drink” and is also the only claim about emotions. Thereby, it is unique from a language perspective (far away from other points in the encoding), but perhaps not unique to consumers, as there might be similar claims on the market already. Also, uniqueness is highly correlated ($r = 0.88$) with the overall rating of claims, some of this correlation could be due to consumers viewing uniqueness as a proxy for “satisfaction”. Lastly, the researched advertising claims are self-selected, and perhaps more unique then the average claim that is on the market already. This might lead to a form of Simpson’s paradox (Sprenger and Weinberger 2021), wherein the population there is a positive correlation between uniqueness and unique language, but perhaps not in this special sub-sample of the data.



(a) Generative Summaries (Factor Model)



(b) BERT

Figure 15: Embedding of the yogurt claims, colored by appeal rating.

Figure 15 again shows the encoding space, but this time colored by the overall rating of the claims and split by the country for which the market was researched. The same claims can get different ratings by country, the reasons for this could lie in what consumers are used to from the domestic market and in cultural differences. For both our proposed method and the BERT-based visualization, we find that claims with similar appeal ratings cluster together. For our document summaries, higher ratings occurr in the northeast and lower ratings towards the southwest of the graph. The clustering of similarly rated claims could be due to these claims being similar in language. For marketers, there are two insights from this: One, getting the overall theme of a claim right, can ensure that customers' perception of this claim is within a certain ballpark. Two, after identifying a fruitful theme for the advertising claim, it is still useful to explore this neighborhood in more fine-grained steps, as locally, there might be small alterations that have large effects on the perception: See e.g. claims 21 (rating in top 4% percentile) and 14 (rating in top 12%), even though these claims are very close to each other. When we consider these observations together with the identified word features and themes from above, it appears that advertising claims for yogurt drinks, which mention the immune system and do not play into a morning theme are appealing to consumers, however, these claims are not perceived as the most unique. Claims that mention the word "natural" are among the most unique claims, again not mentioning a morning theme.

Yoghurt-claims and design motivations (Application 2)

For the yogurt claims, we estimate a 20-factor model and train it until we reach a likelihood of 0.99. We train this model on all unique advertising claims in this dataset. Since the encoding space is of dimension 20, we cannot use it for visualizations directly. Instead, we apply Principal Component Analysis to these 20 factors and extract the first two principal components. The intuition behind this is, that if we capture relevant information with these 20 factors, then we might also pick up on this information with the principal components of these factors. As a comparison, we again apply BERT's CLS token to the same advertising claims and perform PCA on the resulting embedding.

As a first step, we are interested in the location of claims that mention a specific country in the principal component space. An example of such a claim would be: "Yogurt fermented in Germany". These types of claims should have a similar effect on consumers in the domestic market, and hence a good representation of these advertising claims should cluster the different "country-versions" of the same advertising claims together. For this, we flag claims if they mention a specific country and then group these claims into "Claim Types". Some claims don't mention a specific country but rather a local origin. We collect these claims under the flag "Local".

We find three claim types: "Support", "Ingredients", "Origin", and "Made in X". These four claim types are identical in wording and only differ by the mentioned country. The first one is about support for "local" producers, the second one is about domestic ingredients, the third one is about the origin of the product and the fourth one uses the phrase "made in (country)".

In Figure 16 we show the advertising claims in the principal component space and color claims by their mentioned country and adjust the marker type by claim type. For the BERT embedding, we see that the different claim types cluster together across all groups. For the "Ingredients" and "Made in X" types, the claim from the "Local" location is a bit removed from the cluster. This is plausible, as the wording of these claims deviates grammatically from that of the respective claims for the nations. We find a similar result for the PCA visualization of our 20-factor model. However, some of the clusters are not as clearly separated from each other (e.g. "Support" & "Origin"). It appears that both types of document summaries capture the information in the texts in such a way, that similar claims cluster together in a two-dimensional principal component space.

We expand our analysis by investigating whether we can identify the design motivations of the advertising claims again in a two-dimensional principal component space. In Figure 17, we show the first two principal components of the generative summary and the BERT summary for all advertising claims from the five major categories "Packaging", "Local & Responsible", "Sourcing", "Naturality", and "Sustainability", which are labels provided in the data. In the PCA visualization of the yogurt claims based on the BERT embeddings, we see that claims with the "Packaging" theme are separated from the other types of claims in the northwest corner of the plot. The claims with the themes of "Local & Responsible", "Sourcing", "Sustainability", and "Naturality" do not seem to exhibit a specific pattern and are meshed together in the southwest and east areas of the plot.

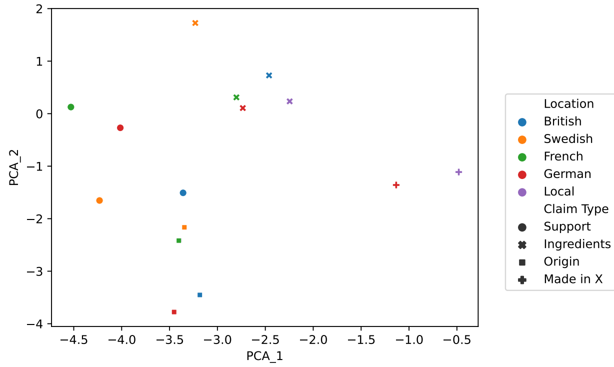
The visualization is different for the generative summaries. Here, advertising claims with the theme "Naturality" follow the second principal component, while the "Local & Responsible", "Sustainability", and the "Packaging" themes follow the first principal component. It also appears that there is a transition between the "Local & Responsible" and "Packaging" themes along this axis.

The theme "Sourcing" appears to be situated at the intersection of "Local & Responsible" and "Naturality." The three "Sourcing" claims, with a first principal component between -4 and -2, and a second principal component between 0 and 2, all have the same phrasing which emphasizes the local origin of the yogurt. The only difference between these three claims is that they use a different "country" to describe the origin (e.g. "*Fermented with just French ingredients*"). We observe a similar pattern for the four southwest "Sourcing" claims, which only differ by an adjective that specifies the origin of the farmers from which the product is sourced. The two most south "Naturaliy" claims are claims that mention the local and organic ingredients of the yogurt. This ties in with the surrounding claims, which are part of either the "Sourcing" or "Local & Responsible" themes. The two "Packaging" claims, which have a first principal component of smaller than one, emphasize how the packaging is not wasteful. The most "northern" packaging claim, i.e. the one with the highest second principal component, points to "plant-based" (i.e. "natural") packaging materials. Since most of these "Packaging" claims revolve around e.g. recyclable or eco-friendly packaging, it fits that most of the "Sustainability" claims are also located among the "Packaging" claims.

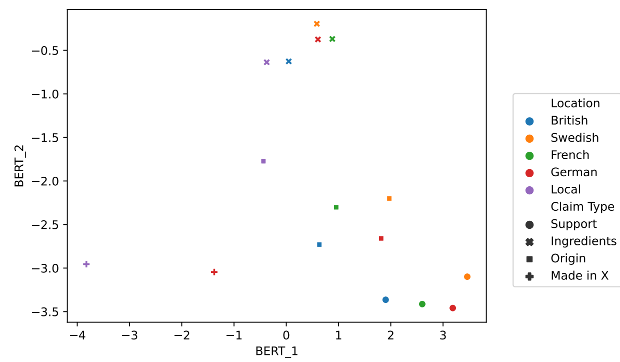
In the word embedding literature, a prominent example to explore the information content of these embeddings is to compare linear combinations of word embeddings to other word embeddings. Mikolov, Ilya Sutskever, et al. (2013) present the following example: The word embedding of the word "Queen" is the closest word embedding to the combination of the word embedding for "King", minus the word embedding for "Man", plus the word embedding for "Woman" (see illustration Equation 3). Here, we try something similar with the advertising claims: We form linear combinations of the generative summaries and generate a text from the resulting vector. For example, we can alter the country of origin for an advertising claim, as follows: The generative summary for "Yoghurt produced in Sweden" - "Swedish ingredients" + "UK ingredients" generates the text "Yoghurt produced in the UK", which is another advertising claim from the training data (see illustration Equation 4). Similarly, in an attempt to generate the claim "Yoghurt produced by local farmer" (which is not part of the training data), we form the combination "Yoghurt produced in Germany" - "German ingredients" + "Local Product", which yields: "Producing farmer Produced by the local farmer [repetition]". While the generated text is not grammatically correct, does not end with an eos-token and contains a repetition, it does seem to capture that we want an advertising claim that emphasizes the local producer of the product, and contains no mention of the ingredients themselves. Finally, we find indications that generated claims make use of associations that the LLM has learned, which go beyond grammatical structures. The combination "Vegetarian ingredients" + "Local production" - "Overpackaging" generates the text "Locally sourced in the United States of America. We are in the process of moving to the United States of America. [repetiton]". While this is not a valid advertising claim, it contains mentions of local production. Perhaps, subtracting the generative summary for "Overpackaging" leads to the generation of "United States of America", which is a country that is never mentioned in the training data, but might be associated with packaging waste in the pre-training data of the used LLM.

$$\text{King} - \text{Man} + \text{Woman} \approx \text{Queen} \tag{3}$$

$$\text{LLM}\left(\mathbf{S}_{\text{Yoghurt produced in Sweden}} - \mathbf{S}_{\text{Swedish ingredients}} + \mathbf{S}_{\text{UK ingredients}}\right) = \text{"Yoghurt produced in the UK"} \tag{4}$$

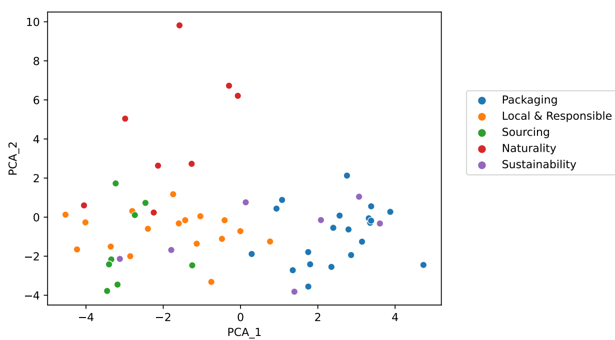


(a) First two principal components of the 20-factor model

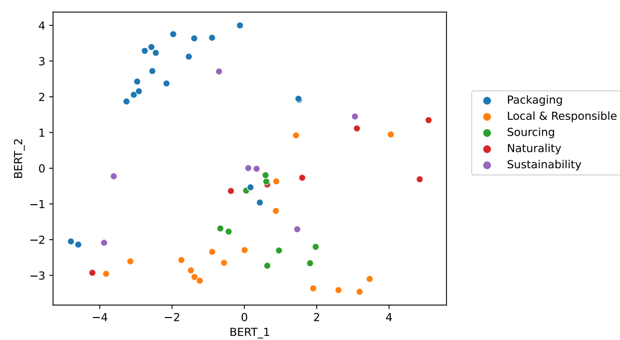


(b) First two principal components of the BERT model

Figure 16: Coloring of claims by whether they mention a certain origin country.



(a) First two principal components of the 20-factor model



(b) First two principal components of the BERT model

Figure 17: Embedding of the yogurt claims, colored by word features.

Managerial Implications

Managers can use these generative summaries to augment the design process of advertising claims. The work by Burnap, Hauser, and Timoshenko (2023) and Ludwig and Mullainathan (2023) propose a framework for guided generation of images. The researchers model an embedding space for images, that fulfills two requirements: One, points in this embedding space can generate new images and two, they form useful features for a supervised machine learning task. For our setting of text data, we also estimate an embedding that can generate new text data and captures inherent information of the advertising claims, which could be used for classification. The frameworks of Burnap, Hauser, and Timoshenko (2023) and Ludwig and Mullainathan (2023) use three components: The encoder, which projects data points into an encoding space (our factor model), the generator (our LLM), which turns points from the encoding space into new data points, and the predictor, which predicts a certain label based on a point from the encoding space (not implemented). In a similar way to Burnap, Hauser, and Timoshenko (2023), businesses could improve their advertising claims, e.g. by training the generator and predictor model based on their market research data, e.g. predicting overall appeal for a claim. We can combine such a predictor model with the low-dimensional factor space of the generative summaries, to guide the exploration of new claims. For this, we only need the decoder part of our factor model: For every point in the factor space, we can get the corresponding generative summary and predict the respective outcome for this generative summary with the predictor model. Through gradient descent, we can now find the points in the factor space that maximizes this predicted outcome. In a classification setting, we can use this technique, e.g. to find the closest point in the factor space which changes the predicted class of the advertising claim.

A framework of generator and predictor model can also help in the design of market research itself. Ludwig and Mullainathan (2023) propose a workflow for generating research hypotheses based on such a model, by finding small steps in the encoding space that maximally change the prediction of the predictor model. They show that these steps yield interpretable and novel hypotheses for their research setting. In a similar vein, marketers might look for a minimal change to an advertising claim, such that this claim changes from being "tangible" to being "intangible". This could help to identify the effects of certain design motivations more clearly, as it operationalizes the "ceteris paribus" principle, by making the smallest possible alteration for this change in class. Typically, such alteration is difficult to obtain in research designs for flexible modalities such as text.

Managers can also explore the factor space to find new design motivations. Figure 12 shows design motivations, which were not designated as themes in the market research study. By looking at the structure of the factor space, and comparing neighbouring claims for the similarities and differences, managers can identify linguistic features, such as the use of certain words of themes.

Discussion

In this research, we propose a novel type of document summary, which maximizies the probability to re-generate the focal document. We show that it requires a form of regularization in practice to capture useful features. We also show that these generative summaries let the generation distribution collapse at the desired target sequence. We can interpolate between two document summaries, and a low-dimensional factor representation captures linguistic similarity of documents. Since this is a new method, further validation is required. Below, we discuss some limitations and challenges of our document summaries, issues with our applications, and end with next steps in this research.

Here, we summarize the results for our four *research objectives*:

1) Yes, we capture meaningful features with the generative summaries, that we can use for classification.
2) Yes, we can re-generate the focal document and it appears that the generative distribution collapses at the target sequence.
3) We can interpolate between two document summaries, and the low-dimensional factor representation appears to make sensible generations, as e.g. linear combinations of generative summaries have meaning. However, the quality of the generations is limited, as many of the generated advertising claims are not intelligible.
4) Yes, we can find marketing insights with our generative summaries, which are relevant and can lead to managerial implications. However, further validation and experiments are required, especially to validate the use of their text generation abilities.

Limitations

Newly generated advertising claims are often unintelligible and there appears to be a lot of whitespace in the factor space. We find three reasons for this. One, we use the GPT-2 model in this paper, which is far from state-of-the-art, e.g. on Huggingface's Open LLM Leaderboard, the best versions of GPT-2 achieve around half the score of the leaderboard leader, which is a model based on Llama-3[16]. To address this problem, we can use a more capable open-source LLM, such as Llama 3 instead, which comes at a larger computational cost. The second reason lies in the construction of our factor space. According to Goodfellow, Bengio, and Courville (2016), factor models can learn and represent features of the data well. However, they struggle with the generation of new data points, which in practice, tend to be mixtures of the learned features rather than realistic data points. This is a pattern we see without newly generated advertising claims, e.g. in Table 5. The third reason is also related to the architecture of our current model but with respect to the number of layers and activation functions. Currently, we a searching for a representation

[16] https://llama.meta.com/llama3/

of the advertising claims in 2 dimensions, and take linear combinations of the resulting factors to create our generative summaries. Perhaps, this structure is too simplistic to represent this language domain well. Introducing additional layers and non-linear activation functions, such as the Swish function (Ramachandran, Zoph, and Le 2017), might yield a representation of the space that is better suited for generation, as we learn the manifold on which these advertising claims live, better. We also observe that more factors yield a more powerful generative summary, however that an estimation of the generative summaries without a factor structure makes them overfit. Currently, there is no data-driven method to regularize the generative summaries, which leaves the choice of the number of factors to the researcher.

There are some challenges, which we inherit from LLMs themselves, such as limited language modeling capabilities in non-English languages and safety and copyright issues when generating new text. However, these are problems that we can circumvent, or at least mitigate, by using a more capable LLM with safeguards, or an LLM that has been trained in a specific language if we are working with non-English texts.

We also want to discuss some observations in the computation of these document summaries. The computation time of our factor model depends on two aspects. The first aspect is the length of the supplied documents and the second is the number of factors. Models that have fewer factors get more and more difficult to compute, especially if we are using more and more claims. The intuition for this is, that it gets difficult to “squeeze” many different documents through only a few factors, while still regenerating each document. One way how this affects the optimization is, that the log-likelihood plateaus and only very slowly increases (if at all). While this might be less of a problem, and more of a statistical fact (we cannot represent certain complexities with a low-dimensional factor model with an arbitrarily high likelihood), it can be an issue in certain applications, e.g. when the researcher needs a specific (low-dimensional) factor representation for a large number of claims.

Our empirical application has low power and does not allow for generalizations, as it is based on a sample of at most 215 observations, with even fewer unique claims. To validate whether our factor model indeed has regions where claims exist that have e.g. a higher appeal rating, we would need to generate new claims from this region and investigate them in an experiment together with the existing claims, as our interpretations are purely correlational.

Expansion of this research

We identify three avenues for future research based on our generative summaries.

The first avenue is with respect to the summary’s generative capabilities. Here, we want to research two extensions, that could improve the quality of its generated text. The first one is to replace the factor mode by a non-linear autoencoder which takes a different document summary, e.g. BERT’s CLS token (Devlin et al. 2018), as an input and learns a mapping from these descriptive embeddings to our generative summaries. The motivation for this is, that non-linear autoencoders appear to be better suited for generation than factor models (see Goodfellow, Bengio, and Courville 2016). A second extension to improve the quality of our text generation could be to introduce an adversary model, creating an adversarial structure. An adversarial structure consists of two models: A generator and an adversary. The generator creates new data points, while the adversary tries to predict whether a data point is genuine or has been generated (Goodfellow et al. 2020). This leads to a game between these two models, where the generator learns to represent the underlying data distribution, such that the adversary cannot distinguish between genuine and generated datapoints anymore. Training such an adversary can also be a way to evaluate the quality of the generated text itself, which is an ongoing research problem (Tatsunori Hashimoto et al. 2019). Exploring how we can reconcile such an adversarial structure with our current training objective of maximizing the generation probability of a focal sequence, is an issue we leave for future research.

The second avenue is the selection of an adequate number of factors. The number of factors is a hyperparameter that the researcher must set when using this model. For the number of factors, there are two interesting research questions. The first one is, to explore how different numbers of factors capture different levels of abstraction in the text data. The second one is, to develop a data-driven method to choose the number of factors. While there can be theoretical motivations for the number of factors (e.g. having two classes as in the test case of intangible and tangible claims), developing a data-driven method would be helpful. Such a method could work by penalizing the use of more factors while rewarding a high likelihood. Perhaps, adapting an information criterion that is based on the likelihood, such as the Bayesian Information Criterion (BIC), could be the starting point for such a development.

Can we adapt the LLM further to a specific domain? For example, the approach by Khattab et al. (2023) also allows for fine-tuning of the LLM with respect to a specific performance metric. Rather than using an out of the box LLM, which has been trained on large text corpora, we could further train and adapt the LLM to the advertising claims, which are special in their use of words and short. Adapting the LLM to the language domain might prove to helpful, especially when we are interested in the factor space of a specific domain. Furthermore, such an adaptation might also improve the quality of new generated texts, as the LLM pronounces sequences of words that sounds like advertising claims more strongly. Besides fine-tuning, Ouyang et al. (2022) argue that especially Reinforcement Learning for Human Feedback (RL-HF), improves the capabilities of an LLM. Augmenting the training of generative summaries by RL-HF, in order to regularize the fit of these embeddings by ensuring that they capture signal rather than noise, is another research avenue to purse in the future.

Conclusion

In this research, we propose a novel, optimal, type of document summary, which maximizes the likelihood of an LLM to generate the document that it summarizes. These document summaries are the only ones, that are numeric and can be used for the generation of new text. In an application to synthetic advertising claims, we show that these document summaries capture relevant information about the documents, but need a factor structure prevent overfitting. We also propose a form of factor model to estimate these document summaries, which yields an interpretable factor space. In an application to market research data, we show that this factor space captures linguistic features and can provide insights on consumer ratings of uniqueness and overall appeal, as well as help with the discovery of design principles. We observe that linguistic uniqueness is not the same as perceived uniqueness by consumers and that claims of similar appeal ratings cluster together. However, we also observe that small adjustments to the wording of claims can lead to relatively large differences in rating. We also show that linear combinations of our generative summaries can be meaningful. We sketch a framework that can help business to adapt their text-based marketing measures, without having to rely on prompt engineering. Open issues of our approach are to solve the problem of selecting the number of factors in a more general way and to improve the quality of newly generated text.

Appendix

Table 6: Tangible (T) and Intangible (I) advertising claims for hair shampoo products. (Created by ChatGPT)

Number	Claim
0 (T)	Transforms dull hair into vibrant, eye-catching color in just one use.
1 (T)	Infuses hair with rich, lasting color that doesn't fade after multiple washes.
2 (T)	Provides 3X more vibrant color compared to leading brands.
3 (T)	Boosts color intensity for up to 8 weeks with regular use.
4 (T)	Enhances natural hair color with luminous, multi-dimensional shades.
5 (T)	Leaves hair visibly shinier and smoother with every application.
6 (T)	Protects color-treated hair from fading and dullness.
7 (T)	Locks in color and moisture for a soft, silky finish.
8 (T)	Revives faded hair color in just 5 minutes.
9 (T)	Strengthens hair while delivering vivid, long-lasting color.
10 (I)	Unleash your most vibrant self with our haircare product.
11 (I)	Experience the magic of bold, beautiful hair color.
12 (I)	Let your hair reflect your inner radiance.
13 (I)	Feel the confidence of truly vibrant hair.
14 (I)	Embrace your colorful side with every wash.
15 (I)	Discover the joy of brilliantly colored hair.
16 (I)	Transform your hair into a work of art.
17 (I)	Reveal the true potential of your hair's color.
18 (I)	Express your personality through stunning hair color.
19 (I)	Let your hair be a reflection of your vibrant spirit.

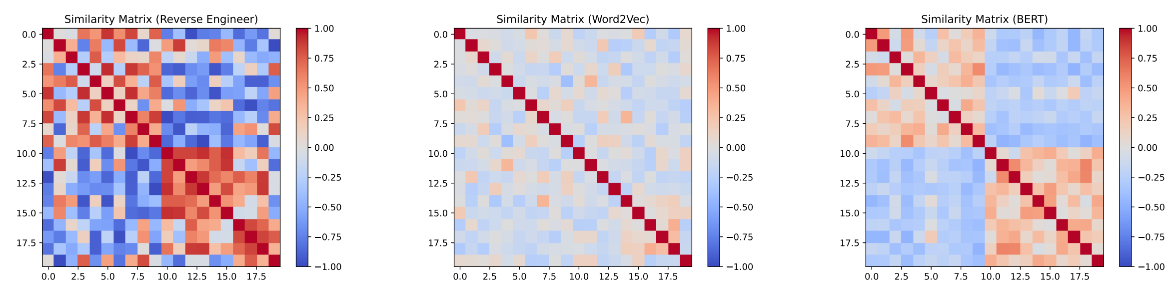
Table 7: Tangible (T) and Intangible (I) advertising claims for hair shampoo products. (Created by ChatGPT)

Number	Claim
0 (T)	Reduces hair breakage by up to 75% in just 4 weeks.
1 (T)	Increases hair strength by 50% after 3 uses.
2 (T)	Visibly smoother hair after one application.
3 (T)	Improves hair thickness by 30% within a month.
4 (T)	Reduces split ends by 60% with regular use.
5 (T)	Enhances hair shine by 40% in 2 weeks.
6 (T)	Reduces frizz by 80% in high humidity conditions.
7 (T)	Promotes 2x faster hair growth.
8 (T)	Decreases scalp dryness by 50% after first use.
9 (T)	Improves hair elasticity by 45% with continuous use.
10 (I)	Transforms your hair's overall vitality.
11 (I)	Revitalizes dull, lifeless hair.
12 (I)	Restores natural hair vibrancy.
13 (I)	Nourishes your hair from root to tip.
14 (I)	Gives your hair a healthy, radiant glow.
15 (I)	Revives the natural beauty of your hair.
16 (I)	Infuses your hair with deep moisture.
17 (I)	Enhances the natural texture of your hair.
18 (I)	Makes hair feel luxuriously soft and silky.
19 (I)	Elevates your hair's natural brilliance.

Number	Claim

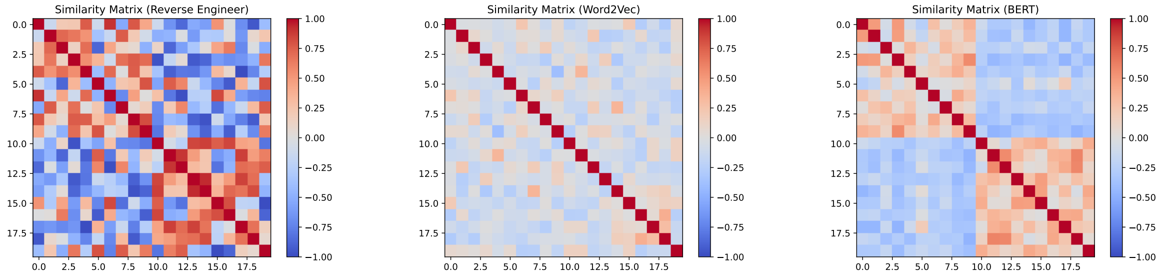
Table 8: Tangible (T) and Intangible (I) advertising claims for surface cleaner. (Created by ChatGPT)

Number	Claim
0 (T)	Removes 99% of all dirt and grime in one wipe.
1 (T)	Restores surfaces to their original shine in seconds.
2 (T)	Leaves no streaks, just a sparkling clean finish.
3 (T)	Eliminates tough stains with ease.
4 (T)	Provides a long-lasting protective shine.
5 (T)	Fast-acting formula, see results immediately.
6 (T)	Safe for all surfaces including glass, metal, and wood.
7 (T)	Dries quickly, leaving surfaces spotless and gleaming.
8 (T)	Contains no harsh chemicals, gentle yet effective.
9 (T)	Reduces cleaning time by half with superior efficiency.
10 (I)	Experience a new level of clean.
11 (I)	Transform your home with a radiant shine.
12 (I)	Let your surfaces shine like never before.
13 (I)	Make every room sparkle with brilliance.
14 (I)	Enjoy the confidence of a spotless home.
15 (I)	Feel the freshness in every corner.
16 (I)	Unlock the secret to dazzling surfaces.
17 (I)	Embrace the power of effortless shine.
18 (I)	Rediscover the beauty of your surfaces.
19 (I)	Breathe easy in a home that gleams.



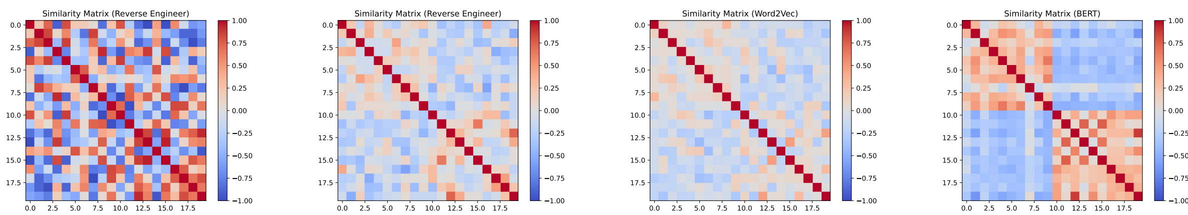
(a) RE (b) W2V (c) BERT

Figure 18: Correlation matrices for the surface cleaner claims.



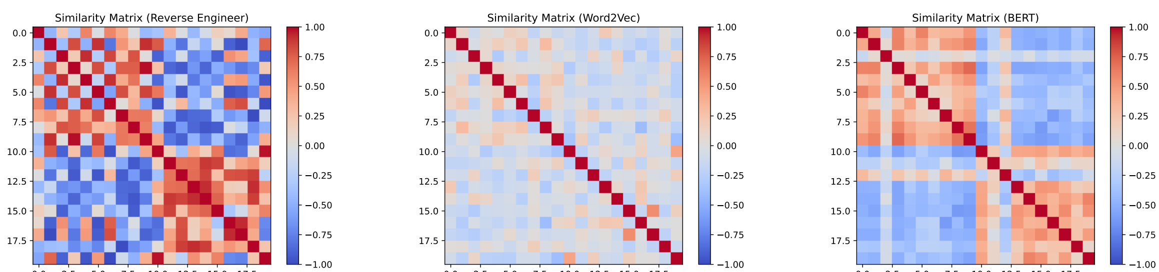
(a) Scrambled order (b) W2V (c) BERT

Figure 19: Correlation matrices for the scrambled surface cleaner claims.



(a) 2-factors (b) 15-factors (c) W2V (d) BERT

Figure 20: Correlation matrices for the hair claims with color attribute.



(a) RE (b) W2V (c) BERT

Figure 21: Correlation matrices for the hair color with health attribute.

References

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. "Layer Normalization." arXiv. https://arxiv.org/abs/1607.06450.

Berger, Jonah, Ashlee Humphreys, Stephan Ludwig, Wendy W. Moe, Oded Netzer, and David A. Schweidel. 2020. "Uniting the Tribes: Using Text for Marketing Insight." *Journal of Marketing* 84 (1): 1–25. https://doi.org/10.1177/0022242919873106.

Burnap, Alex, John R Hauser, and Artem Timoshenko. 2023. "Product Aesthetic Design: A Machine Learning Augmentation." *Marketing Science* 42 (6): 1029–56.

Chakraborty, Shayak, and Partha Pakray. 2024. "Abstractive Summarization Evaluation for Prompt Engineering." In *Advances in Visual Informatics*, edited by Halimah Badioze Zaman, Peter Robinson, Alan F. Smeaton, Renato Lima De Oliveira, Bo Nørregaard Jørgensen, Timothy K. Shih, Rabiah Abdul Kadir, Ummul Hanan Mohamad, and Mohammad Nazir Ahmad, 14322:629–40. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7339-2_50.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." https://doi.org/10.48550/ARXIV.1810.04805.

Diederik P. Kingma, Diederik P. Kingma, Jimmy Ba, and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv: Learning*, December.

Eggers, Felix, Henrik Sattler, Thorsten Teichert, and Franziska Völckner. 2022. "Choice-Based Conjoint Analysis." In *Handbook of Market Research*, edited by Christian Homburg, Martin Klarmann, and Arnd Vomberg, 781–819. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-57413-4_23.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, Massachusetts: The MIT Press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. "Generative Adversarial Networks." *Communications of the ACM* 63 (11): 139–44. https://doi.org/10.1145/3422622.

Hartmann, Jochen, and Oded Netzer. 2023. "Natural Language Processing in Marketing." In *Artificial Intelligence in Marketing*, edited by K. Sudhir and Olivier Toubia, 20:191–215. Review of Marketing Research. Emerald Publishing Limited. https://doi.org/10.1108/S1548-643520230000020011.

Huang, Zihao, and Tao Chen. n.d. "Enhanced News Summarization Using Large Language Models and Advanced Prompt Engineering."

Ioffe, Sergey, and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." arXiv. https://doi.org/10.48550/arXiv.1502.03167.

Jeffrey Pennington, Jeffrey Pennington, Richard Socher, Richard Socher, Christopher Manning, and Christopher D. Manning. 2014. "Glove: Global Vectors for Word Representation," October, 1532–43. https://doi.org/10.3115/v1/d14-1162.

Khattab, Omar, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, et al. 2023. "DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines." arXiv. https://arxiv.org/abs/2310.03714.

Li, Cheng, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023. "Large Language Models Understand and Can Be Enhanced by Emotional Stimuli." arXiv. https://arxiv.org/abs/2307.11760.

Liu, Nelson F., Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. "Lost in the Middle: How Language Models Use Long Contexts." arXiv. https://doi.org/10.48550/arXiv.2307.03172.

Lu, Sheng, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. "Are Emergent Abilities in Large Language Models Just In-Context Learning?" arXiv. https://doi.org/10.48550/arXiv.2309.01809.

Ludwig, Jens, and Sendhil Mullainathan. 2023. "Machine Learning as a Tool for Hypothesis Generation." w31017. Cambridge, MA: National Bureau of Economic Research. https://doi.org/10.3386/w31017.

Mikolov, Tomas, Ilya Sutskever, Ilya Sutskever, Kai Chen, Kai Chen, Kai Chen, Greg S. Corrado, Greg S. Corrado, Jeff Dean, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and

Their Compositionality” 26 (December): 3111–19.

Mikolov, Tomas, Kai Chen, Kai Chen, Kai Chen, Greg S. Corrado, Greg S. Corrado, J. Michael Dean, and Jeffrey Dean. 2013. “Efficient Estimation of Word Representations in Vector Space,” January.

Onan, Aytuğ. 2023. “Hierarchical Graph-Based Text Classification Framework with Contextual Node Embedding and BERT-based Dynamic Fusion.” *Journal of King Saud University - Computer and Information Sciences* 35 (7): 101610. https://doi.org/10.1016/j.jksuci.2023.101610.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. “Training Language Models to Follow Instructions with Human Feedback.” *Advances in Neural Information Processing Systems* 35 (December): 27730–44.

Pearson, Karl. 1901. “LIII. *On Lines and Planes of Closest Fit to Systems of Points in Space*.” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (11): 559–72. https://doi.org/10.1080/14786440109462720.

Phoenix, J., and M. Taylor. 2024. *Prompt Engineering for Generative AI*. O'Reilly Media.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. “Improving Language Understanding by Generative Pre-Training.”

Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. 2017. “Searching for Activation Functions.” arXiv. https://arxiv.org/abs/1710.05941.

Shen, Dinghan, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. “Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms.” https://doi.org/10.48550/ARXIV.1805.09843.

Sprenger, Jan, and Naftali Weinberger. 2021. “Simpson's Paradox.”

Tatsunori Hashimoto, Tatsunori B. Hashimoto, Hugh Zhang, Hugh Zhang, Percy Liang, and Percy Liang. 2019. “Unifying Human and Statistical Evaluation for Natural Language Generation.” *arXiv: Computation and Language*, April.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” *Advances in Neural Information Processing Systems* 30.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, et al. n.d. “Emergent Abilities of Large Language Models.”

Yinhan Liu, Yinhan Liu, Myle Ott, Myle Ott, Naman Goyal, Naman Goyal, Jingfei Du, et al. 2019. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” *arXiv: Computation and Language*, July.

Zamfirescu-Pereira, JD, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. “Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts.” In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–21.

Zhang, Jingzhao, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2020. “Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity.” arXiv. https://doi.org/10.48550/arXiv.1905.11881.