Chapter 2: Background

This chapter reviews relevant research for gleaning information from textual data. We start with the background of early topic models (Section 2.1) and present day neural topic models (Section 2.2). We then review two lines of research that inspires this proposal: interactive topic models (Section 2.3) and large language models for information extraction (Section 2.5)

2.1 Probabilistic Models

The fundamental probabilistic model that topic models are still built on today is LDA. LDA assumes that documents are composed of an admixture of topics, where each topic is a distribution over the words. LDA is built upon the Dirichlet distribution. The Dirichlet distribution is a family of continuous multivariate probability distributions parameterized by a vector of positive reals. It is the multivariate generalization of the beta distribution, used extensively in Bayesian statistics and NLP. The Dirichlet distribution, denoted as $Dir(\alpha)$, where $(\alpha = (\alpha_1, \ldots, \alpha_K))$ and each $(\alpha_k > 0)$, is defined over a (K)-dimensional simplex (a vector ($\mathbf{x}$) such that $(\sum_{k=1}^{K} x_k = 1)$ and $(x_k \geq 0) \forall (k))$. The probability density function of a Dirichlet-distributed random variable $(\mathbf{x})$ is given by:

$$f(\mathbf{x}; \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \tag{2.1}$$

where $(B(\alpha))$ is the multinomial beta function, a normalization constant ensuring that the total probability integrates to one. Each parameter (α_k) influences the concentration of the (k^{th}) component (x_k) of the vector. Higher values of (α_k) increase the probability that (x_k) is close to one, indicating a stronger presence or higher probability of the corresponding category or topic. When all (α_k) are equal and greater than one, the distribution is symmetric and peaks at $(1/K)$. In LDA, the Dirichlet distribution is used to model the distributions of topics in documents (document-topic distributions) and the distributions of words in topics (topic-word distributions),

1. Choose $\theta_d \sim$ Dirichlet(α).

2. For each of the (N) words (w_i) in the document:

   (a) Choose a topic $z_i \sim$ Multinomial(θ_d).

   (b) Choose a word $w_i \sim$ Multinomial(ϕ_{z_i}), where (ϕ_{z_i} is the word distribution for topic (z_i).

The joint distribution of the topic mixtures (θ), a set of (N) topics (z), and (N) words (w) in the documents is given by:

$$[p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta),] \tag{2.2}$$

where the goal of LDA is to infer the hidden topic structures that best explain the observed words in the documents.

The goal of LDA is to infer the latent variables (θ, z, and β) given the observed words w in the documents. This involves computing the posterior distribution $p(\theta, z, \beta|w)$, which

is computationally intractable. Approximation techniques such as Gibbs Sampling [47] and Variational Inference [1] are commonly used for inference.

Since these topics are simply a distribution over words, the first step after training is often labeling the topics. Either by selecting top words through a Markov chain Monte Carlo algorithm [47, 48] or through manual generation of descriptive topics [49, 50].

The suite of options for naming topics for probabilistic models go far beyond this. The Bayesian framework, encourages the incorporation of expert knowledge into interactive topic models. This can either represent a dictionary [51], word lists from psychology [52], or the needs of a business organization [20]. This feedback to a model helps match a user's information needs or reflect world knowledge and common sense.

The next step, would be a fully supervised model [10], where every training document has a topic label. But this requires substantially more interaction with the user than giving feedback on a handful of topics.

Probabilistic models are still the go-to in discovering structure in documents collections such as digital humanities [53], bioinformatics [54], political science [55], and social science [56]. But as data has become more complex, probabilistic models are not always sufficient.

Traditional probabilistic topic models, conceptualize documents as mixtures of discrete topics, where each topic is characterized by a distribution over words. This method relies heavily on the bag-of-words assumption, which, although effective for capturing thematic structures, generally ignores the order of words and thus loses the rich contextual information present in text. Moreover, the discrete representation of topics in such models often lacks the flexibility to capture nuanced semantic relationships between words and topics, resulting in a representation that can be less powerful for complex language understanding tasks than the representational

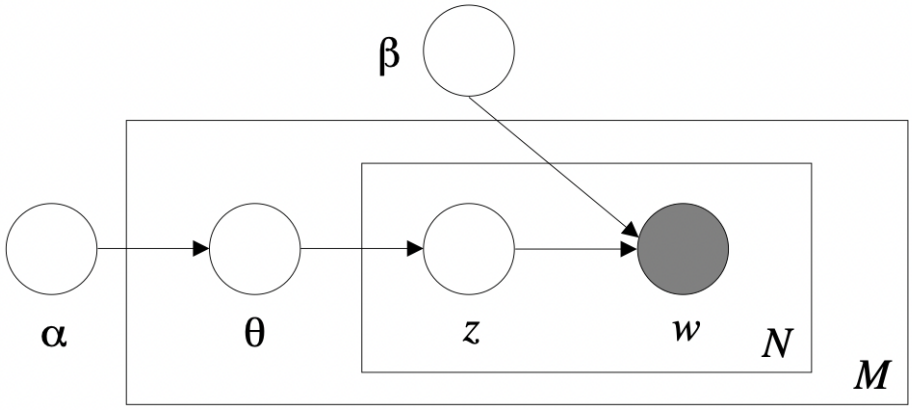

Figure 2.1: Graphical model for the probabilistic LDA model by **(author?)** [1]. The outer plate represents the document and the inner plate, the repeated topics and word for a given document

power of neural networks. Which can learn dense, continuous vector representations of text.

2.2 Neural Topic Models

Neural topic models incorporate deep learning techniques to enhance the discovery and interpretation of topics within text data. These models leverage word embeddings and neural networks to capture complex semantic relationships.

Word embeddings provide dense vector representations of words that capture semantic similarities. Where each word is represented as a constant, dense vector in a predefined D-dimensional space. These vectors, learned through training, encode latent semantic and syntactic features of words based on their contexts. This transformation allows words with similar meanings to have similar vector representations, facilitating complex linguistic computations. For example, Word2Vec's Skip-Gram model [57] optimizes the log probability of observing a context word given a target word:

$$\max \sum_{t=1}^{T} \sum_{-C \leq i \leq C, i \neq 0} \log p(w_{c,i}|w_t), \tag{2.3}$$

where (C) is the context window size, and (T) is the total number of words in the corpus.

One of the most common frameworks for neural topic models uses the Variational Autoeconder (VAE). VAE learn latent representations of data, providing a generative model [58]. In topic modeling, a VAE consists of an encoder and a decoder, connected by a latent topic space. The encoder maps a document (d) to a latent topic distribution (θ), approximating the posterior $(q(\theta|d))$. While the decoder reconstructs the document from (θ), generating a distribution over words and capturing $(p(d|\theta))$.

The VAE is trained by maximizing the Evidence Lower BOund (ELBO):

$$\text{ELBO} = E_{q(\theta|d)}[\log p(d|\theta)] - \text{KL}[q(\theta|d)||p(\theta)], \tag{2.4}$$

where the first term is the expected log-likelihood of the data given the latent topics, and the second term is the Kullback-Leibler divergence.

Much research was focused on adapting VAE's for topic modeling; [59, 60] focus on developing different prior distributions for the reparameterization step of VAE, such as using hybrid stochastic-gradient MCMC and approximating Dirchelt samples with Laplace approximations. VAE-NTM also were extended to work with different architectures, **(author?)** [61] developed a sequential NTM where the model generates documents by sampling a topic for one whole sentence at a time and uses a RNN decoder. **(author?)** [62] uses variational inference to develop embeddings for the actual topics, inducing a distributed representation of the kth topic in the semantic space of words.

While neural topic models are able to learn complex, semantic relationships from text, most neural topic models still result in static topics.

2.3 Interactive Topic Models

Interactive topic models represent a significant evolution in the field of text analysis by incorporating human feedback into the traditional topic modeling process. This development stems from the recognition that while automated algorithms can uncover hidden thematic structures within large datasets, they may not always align with human interpretation or specific analytical needs. Interactive topic modeling addresses this gap, creating a dynamic dialogue between the user and the algorithm to refine and guide the discovery process.

The initial foray into interactive topic modeling was motivated by the need for more interpretable and relevant topic models. Traditional models like LDA provided a foundation for unsupervised topic discovery but lacked mechanisms for incorporating direct user input. Early interactions sought to incorporate rich priors [10], syntactic information [63], or structural priors from covariates [64]. Topics are typically modeled as independent distributions over words. However, in many real-world texts, topics are not entirely independent; they often exhibit thematic overlaps. For instance, topics related to ”health” and ”fitness” might share several keywords. Correlated Topic Models (CTM) [65], extend LDA by using a logistic normal distribution to model topic correlations. This approach allows the model to express and capture the intuition that some sets of topics are more likely to co-occur than others. While not interactive, the logistic normal prior is placed over the distribution of topic proportions, enabling the model to capture complex correlations among topics, and improving the relevancy of topics.

(author?) [66] compared labels generated by users after seeing topic visualizations with automatically generated labels. **(author?)** [20] provides a method for iteratively updating topics by enforcing constraints. **(author?)** [67] make the task of labeling into an optimization problem, to provide an objective probabilistic method for labeling. But there has yet to be work that extends this iterative process to neural-based topic models in an intuitive and natural sense such as I-NTM.

Another significant advancement is the use of visual analytics in interactive topic models. Visualization tools help users understand the complex relationships between words, topics, and documents, enabling more intuitive and effective feedback. These tools often provide real-time updates to the model, allowing users to see the immediate impact of their input. [68]. Even more recently, **(author?)** [69] developed a user-interface for interactive and guided topic modeling, based on Gibbs sampling.

Similar to interactive models, are the anchor topic models, introduced by **(author?)** [70], which leverage the concept of anchor words to simplify the topic discovery process. The key insight behind this method is that if a word is specific enough to a topic, its occurrence can strongly indicate the topic's presence in a document. By focusing on these anchor words, the model can efficiently uncover the underlying thematic structure of a dataset.

There has been extensive work in the area of anchor-based topic modeling—where a single word is used to identify a topic. **(author?)** [71] present "Tandem Anchors" where multi-word anchors are used to interactively guide topics. **(author?)** [72] developed a framework for interactively establishing anchors and alignment across languages. **(author?)** [73] introduces a protocol that allows users to interact with anchor words to build interpretable topic.

2.4 Limitations of Topic Models

Despite their widespread use in text analysis, topic models, both probabilistic and neural, suffer from a range of limitations that restrict their effectiveness in real-world applications. One of the most persistent issues is the tendency of topic models to produce semantically incoherent or redundant topics. For example, classical models such as LDA often output topics whose top words are loosely related or overly generic, which hinders interpretability. While neural models incorporating word embeddings offer modest improvements in coherence, they are still prone to diffuse topics, especially in noisy or domain-specific corpora.

Another limitation stems from the bag-of-words assumption, which ignores word order and syntactic structure. This prevents models from distinguishing between different senses of polysemous words or capturing nuanced semantic relationships. Although word embeddings partially help this by encoding distributional semantics, the models still operate at the level of unordered word co-occurrence, limiting their expressiveness. Moreover, topic models require users to specify the number of topics in advance, a hyperparameter that is rarely obvious. Too few topics results in underfitting and broad, overlapping topics, while too many leads to redundant ones. While nonparametric approaches like the Hierarchical Dirichlet Process (HDP) [74] allow the model to infer the number of topics, they are computationally intensive and difficult to scale.

A further challenge lies in the lack of supervision or user control. Standard topic models are fully unsupervised and offer no mechanism for aligning topics with user-defined categories or domain-relevant terms. This severely limits their utility in applied settings where semantic alignment and task-specific relevance are crucial. Supervised extensions such as sLDA [75] constrain the model with label information, but they reduce flexibility for exploratory tasks. More

recently, interactive topic modeling [76] has aimed to bridge this gap by enabling users to provide constraints during inference, but these techniques remain largely limited to probabilistic models.

Perhaps the most fundamental limitation is that traditional topic models are static and do not adapt to user intent. Once trained, the model cannot be refined, corrected, or steered by user feedback. This makes them poorly suited for dynamic or exploratory analysis scenarios, where interpretability and interactivity are critical. These limitations collectively motivate the development of interactive and neural-interactive topic modeling frameworks, which aim to produce more coherent, adaptable, and user-aligned representations of thematic structure in text.

2.5 Large Language Models

Large Language Models (LLMs) have emerged as a transformative force in natural language processing, enabling high-performance systems across a wide array of tasks with little or no task-specific supervision. This paradigm shift began with the introduction of the Transformer architecture by **(author?)** [3], which replaced recurrent and convolutional structures with self-attention mechanisms. The core attention function computes token-wise contextual relevance:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

where Q, K, and V are the query, key, and value matrices derived from the input, and d_k is the dimensionality of the keys. This formulation allows the model to capture long-range dependencies in a fully parallelizable fashion, making it highly scalable.

Building on this architecture, **(author?)** [77] introduced BERT, a bidirectional encoder trained using a masked language modeling (MLM) objective. By predicting masked tokens

Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %

Figure 2.2: GPT-4 performance on a common tests [2]

given both left and right context, BERT learns rich, contextual representations of language. This pretraining setup enables strong transfer performance across diverse NLP tasks, from question answering to sentiment analysis.

In contrast, the GPT family of models [78, 79] follows an autoregressive approach, training unidirectionally to predict the next word in a sequence:

$$L_{\text{LM}}(\theta) = \sum_{t} \log p(w_t | w_{<t}; \theta) \tag{2.6}$$

where w_t is the t-th word, and $w_{<t}$ represents all preceding words in the sequence.

GPT-3, with 175 billion parameters, demonstrated that scaling model size substantially improves few-shot and even zero-shot capabilities. This finding catalyzed rapid progress in LLM development, leading to models such as PaLM-2 [80], LLaMA-2 [81], and GPT-4 [2], which have been deployed in applications ranging from chatbots [82] and legal text classification [83], to biomedical domain adaptation [84] and scientific literature processing [85].

A key strength of LLMs lies in their ability to produce deeply contextualized embeddings.

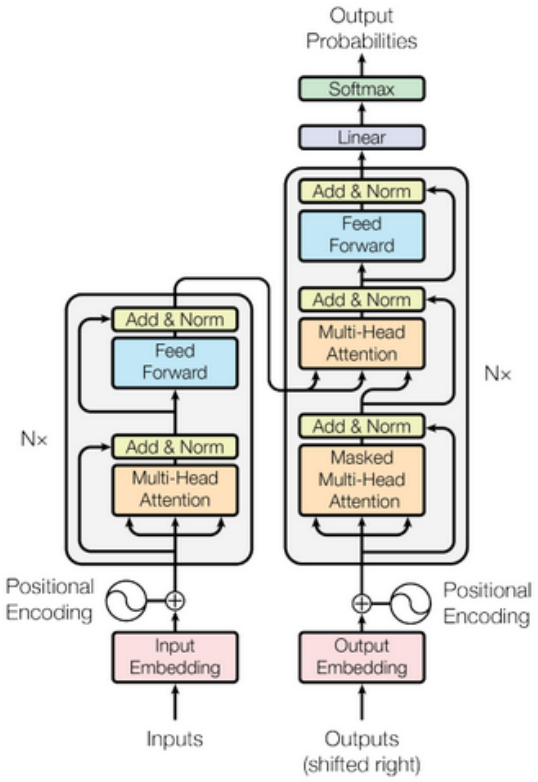

Figure 2.3: Transformer architecture [3]

Trained on massive corpora using self-supervised learning, these models encode fine-grained semantic, syntactic, and even pragmatic information about language. Unlike earlier word embedding methods that assign static vectors to words, LLMs capture how meanings shift across contexts—critical for tasks like topic modeling, text classification, and psychological inference. These representations serve as robust, adaptable features for both supervised and unsupervised downstream tasks, often requiring minimal fine-tuning to perform competitively.

2.5.1 Zero-Shot and Fine-Tuned Models

Large language models have shown exceptional performance on a wide range of text prediction tasks, including sentiment analysis, classification, and personality inference. These models can be used in two primary modes: zero-shot or few-shot inference, and fine-tuning. In the zero-shot

setting, a pretrained LLM is given a task description and input in natural language and asked to generate a prediction without any additional training. Few-shot learning extends this by providing a small number of labeled in-context examples. This approach was popularized by GPT models, which demonstrate that sufficiently large models can generalize to a wide variety of tasks with minimal supervision using only prompt engineering [79]. Further work such as chain-of-thought prompting has shown that careful prompt design can significantly improve reasoning ability even in few-shot settings [86]. However, few-shot performance is often sensitive to prompt wording and ordering, motivating research on prompt calibration to improve stability and reduce variance [87].

Fine-tuning, by contrast, adapts a pretrained model to a specific downstream task using supervised training. In this approach, a classification head is typically added to the model and trained using labeled data, updating either the full model parameters or a small subset. Early work like ULMFiT [88] and BERT [77] demonstrated the effectiveness of task-specific fine-tuning for a broad range of NLP applications. More recently, parameter-efficient methods such as LoRA [89] have enabled fine-tuning of large-scale models using only a fraction of the original parameters, making it feasible to adapt LLMs even in resource-constrained environments.

In this dissertation, LLMs are used in both zero-shot and fine-tuned modes to predict psychological dispositions from user-generated text. While LLMs outperform classical baselines in predictive accuracy, especially in low-data settings, this task reveals deeper limitations. Predictions are often based on superficial stylistic features rather than genuine psychological cues, and performance can vary significantly depending on input phrasing or domain mismatch. Moreover, evaluation remains a fundamental challenge: standard classification metrics fail to capture the complexity of predicting latent human traits. These findings underscore that, despite their capabilities,

LLMs require structured human feedback, careful prompt design, and principled evaluation to ensure reliability and interpretability in high-stakes or subjective applications. We will see in Section 4.1 how these models can predict psychological dispositions.