**Reading 7**

**Topic Modeling , Semantic Analysis, Sentiment Analysis, Covariance**

**Visualizing topic models with scatterpies and t-sne**

For a computer to understand written natural language, it needs to understand the symbolic structures behind the text.

Using some of the NLP techniques and can enable a computer to classify a body of text and can explain more about themes etc. Below is some of them:

1. Topic modeling (themes, clustering)
2. Word/phrase frequency (and “keyword searching”)
3. Text visualization
4. Embeddings
5. Sentiment analysis (positive/negative, subjective/objective, emotion-tagging)
6. Text similarity (e.g. “cosine similarity”)TF-IDF (term frequency/inverse document frequency)
7. Part-of-speech tagging

**Topic modeling:** An enhanced LDA model that uses sampling to resemble a semi-supervised approach rather than an unsupervised one. The output from the topic model is a document-topic matrix

To reduce *T* topics into a easily-visualizable 2-dimensional space we can use t-Distributed Stochastic Neighbor Embedding (or t-SNE).

**Latent Dirichlet allocation (LDA):**

LDA is a 3-level hierarchical Bayesian model, with each item of a collection modeled as a finite mixture over an underlying set of topics. Each topic is, modeled as an infinite mixture over an underlying set of topic probabilities. It is a generative probabilistic model for collections of discrete data such as text corpora.

A **word** is the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1,...,V}. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the vth word in the vocabulary is represented by a V-vector w such that wv = 1 and wu = 0 for u 6= v.

A **document** is a sequence of N words denoted by w = (w1,w2,...,wN), where wn is the nth word in the sequence.

A **corpus** is a collection of M documents denoted by D = { w1,w2,...,wM}.

The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.1 LDA assumes the following generative process for each document w in a corpus D: 1. Choose N ∼ Poisson(ξ). 2. Choose θ ∼ Dir(α). 3. For each of the N words wn: (a) Choose a topic zn ∼ Multinomial(θ). (b) Choose a word wn from p(wn | zn,β), a multinomial probability conditioned on the topic zn .

In tf-idf scheme, for each document in the corpus, a count is formed of the number of occurrences of each word. After suitable normalization, this term frequency count is compared to an inverse document frequency count, which measures the number of occurrences. The end result is a term-by-document matrix X whose columns contain the tf-idf values for each of the documents in the corpus. Thus the tf-idf scheme reduces documents of arbitrary length to fixed-length lists of numbers.

**Latent semantic indexing (LSI)** uses a singular value decomposition of the X matrix to identify a linear subspace in the space of tf-idf features that captures most of the variance in the collection. This approach can achieve significant compression in large collections. LSI uses a singular value decomposition of the X matrix to identify a linear subspace in the space of tf-idf features that captures most of the variance in the collection. This approach can achieve significant compression in large collections. Furthermore, the derived features of LSI, which are linear combinations of the original tf-idf features, can capture some aspects of basic linguistic notions such as synonymy and polysemy.

**Probabilistic LSI (pLSI)** model, also known as the aspect model, as an alternative to LSI. The pLSI approach, which we describe in detail in Section 4.3, models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of “topics.” Thus each word is generated from a single topic, and different words in a document may be generated from different topics. Each document is represented as a list of mixing proportions for these mixture components and thereby reduced to a probability distribution on a fixed set of topics. This distribution is the “reduced description” associated with the document.

**Semantic analysis** describes the process of understanding natural language;the way that humans communicate–based on meaning and context.

The semantic analysis of natural language content starts by reading all of the words in content to capture the real meaning of any text. It identifies the text elements and assigns them to their logical and grammatical role. It analyzes context in the surrounding text and it analyzes the text structure to accurately disambiguate the proper meaning of words that have more than one definition.

### **Semantic Classification Models**

Topic classification: sorting text into predefined categories based on its content.

Sentiment analysis: detecting positive, negative, or neutral emotions in a text to denote urgency. For example, tagging Twitter mentions by sentiment to get a sense of how customers feel about your brand, and being able to identify disgruntled customers in real time.

Intent classification: classifying text based on what customers want to do next. You can use this to tag sales emails as “Interested” and “Not Interested” to proactively reach out to those who may want to try your product.

Covariance is a measure of how changes in one variable are associated with changes in a second variable. Specifically, covariance measures the degree to which two variables are linearly associated. However, it is also often used informally as a general measure of how monotonically related two variables are.

Probabilistic grammars are attractive for several reasons. Like symbolic grammars, they are amenable to inspection by humans, so that it is relatively easy to understand what tendencies the model has captured if the underlying rules are understandable

they model frequency and provide a mechanism for reasoning in the face of ambiguity, which is ubiquitous in natural language. NLP applications of probabilistic grammars and their generalizations include parsing, machine translation and question answering.

The modeling of covariance among the probabilities of grammar derivation events, and propose the use of logistic normal distributions over multinomials to build priors over grammars. The observation that various grammar parameters are expected to be correlated because of the elements in language they represent share linguistic properties. Noting that grammars are built out of a large collection of multinomials, we introduce shared logistic normal distributions to allow arbitrary covariance among any grammar probabilities.

**Fake news detector using machine learning**

There are recent advancements in technology which allows us to handle the problem of fake news using machine learning classifiers. Firstly we train a model to classify neutral vs biased articles titles using classificationbox.

Machine Box provides FakeBox, a fake news classifier trained with significant datasets based on common sense classification of news articles.

Preparing training data: Create a folder with subfolders such as   
/biased, /neutral, /satire, /junksci. Consider good number of examples for each topic and provide it with key, content and type and continue to training and based on the accuracy score we can continue predicting results.

**Humility in AI: Building trustworthy and ethical AI systems**

Obviously, a deliberately misleading story is “fake news” but lately blathering social media discourse,  is changing its definition. Some now use the term to dismiss facts counter to their preferred viewpoints, the most prominent example is President Trump

Social media has both perks and cons. It it is used in a positive way, it is low cost, easy access, and rapid dissemination of information allow users to consume and share the news. But it can make viral “fake news”, i.e., low-quality news with intentionally false information.

First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult to detect based on news content. Thus, we need to include auxiliary information, such as user social engagements on social media, to help differentiate it from the true news. Second, exploiting this auxiliary information is nontrivial in and of itself as users’ social engagements with fake news produce data that is big, incomplete, unstructured, and noisy.

Detecting fake news on social media, including two phases: characterization and detection.

Fake news on social media has its unique characteristics. For e.g. malicious accounts can be easily and quickly created to boost the spread of fake news, such as social bots, cyborg users, or trolls.

The aforementioned theories are valuable in guiding research of fake news detection. Existing algorithms for fake news detection can be generally categorized as (i) News Content Based and (ii) Social Context Based.

News content based approaches focus on extracting various features in fake news content, including knowledge-based and style-based.

Social context based approaches aim to utilize user social engagements as auxiliary information to help detect fake news.

To tackle the challenges due to fake news spread on social media , many existing works exploit various features, from a network perspective, to detect and mitigate fake news. In essence, news dissemination ecosystem involves three dimensions on social media, i.e., a content dimension, a social dimension, and a temporal dimension.

**Data Gathering or Wrangling**

There were two parts to the data acquisition process, getting the “fake news” and getting the real news. The first part was quick, Kaggle released a [fake news dataset](https://www.kaggle.com/mrisdal/fake-news) comprising of 13,000 articles published during the 2016 election cycle.

The second part was… a lot more difficult.  To acquire the real news side of the dataset, to host news and opinion articles from across the political spectrum. Articles on the website are categorized by topic (environment, economy, abortion, etc…) and by political leaning (left, center, and right).

### **Modeling**

Since this is a text classification project, Naive Bayes classifier can be used as a standard for text-based data science projects. The real work in formulating a model was the text transformation (count vectorizer vs tfidf vectorizer) and choosing which type of text to use. In https://www.kdnuggets.com/2017/04/machine-learning-fake-news-accuracy.html, a fake news detection approach is given which followed processes of data wrangling, modeling and obtained an accuracy of 88%.

**Design of the Topic Naming + Cosine Similarity**

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. In this context, the two vectors I am talking about are arrays containing the word counts of two documents.