Text Analysis for Psychology: Methods, Principles, and Practices

Brendan Kennedy[1], Ashwini Ashokkumar[2], Ryan Boyd[3–5], Morteza Dehghani[1,6]

[1] University of Southern California, Dept. of Computer Science, USA

[2] University of Texas at Austin, Dept. of Psychology, USA

[3] Lancaster University, Dept. of Psychology, United Kingdom

[4] Lancaster University, Security Lancaster, United Kingdom

[5] Lancaster University, Data Science Institute, United Kingdom

[6] University of Southern California, Dept. of Psychology, USA

Author Note

We are grateful to Matthias Mehl, Kenji Sagae, Nils Karl Reimer, Aida Mostafazadeh Davani, Alina Arseniev-Koehler, Philip Resnik, and Russell Richie for their help in reviewing this work.

In 1893, the editor of the New York World, John Speed, wanted to see whether newspapers had changed their approach to reporting the news. He suspected that newspapers were beginning to get lured into the financial promises of sensationalism and that such distractions were leading them away from fulfilling their democratic duties. He compared the content of New York dailies published in 1881 and in 1893 (Sumpter, 2001). Rather than looking for anecdotes to support his hypothesis, Speed instead adopted a quantitative approach. He pored over the newspaper articles, categorized them into themes such as science or literature, and then counted the number of articles from each category. He concluded that journalism had reduced coverage of matters relating to science, government, and literature and instead prioritized gossip and scandal. John Speed's project is the first known quantitative content analysis of a newspaper.

Fast-forward over one hundred years, and quantitative language analysis is an undertaking with high speed computers, sophisticated data-driven algorithms for parsing and representing text, domain-specific resources at the intersection of language and psychology, and, most recently, an abundance of naturalistic text datasets full of untapped insight and potential (see Boyd & Schwartz, 2021). This ecosystem can be dizzying to navigate; there are more ways to go wrong than to go right. But the example provided by Speed's analysis of newspaper articles gives some clarity that we can revisit throughout this process. Text analysis in psychology begins and ends with well-articulated questions or ideas about language users. Speed's hypothesis was that newspapers were prioritizing frivolity over their serious duty as democratic carriers of information. The clarity of the idea allowed its straight-forward application through manual coding, counting of newspaper articles, and interpretation of the results. Today, the subject matter that psychologists study may be different, the methods may be more advanced, and the data more diverse, but the modern language analysis process is still supervised by clear ideas and well-founded questions.

Psychology in Language

Language analysis is usually guided by two sets of design considerations. First, the relationship between theory and language is motivated by asking: what theories inspire the current research? What constructs, questions, and hypotheses are suggested by the literature, or by a researcher's own intuitions, interests, and experience? Just as with traditional methods such as surveys and experiments, the researcher must hone in on the constructs they want to study. For example, the large body of scholarship on the multidimensional structure of social class hierarchies (e.g., Chan & Goldthorpe, 2007; DiMaggio, 1982) can motivate research that quantifies dimensions of class distinctions in language (Kozlowski et al., 2019). The second set of considerations concerns the particular ways that the identified construct appears in language, whether it be through patterns in automatically recorded, everyday conversations (Mehl et al., 2001), temporal trends in the cultural indicators in published books (Michel et al., 2011), the frequency of a construct appearing in social media (Mooijman et al., 2018; H. A. Schwartz et al., 2013), or any other setting in which a construct might be identifiable, and data is available. Considering the manifestation of a construct in language can guide analysis directly (i.e., by specifying a target measurement *a priori*), or by clarifying how research questions might be operationalized. These two sets of considerations connect theories and language data, driving hypothesis formation, data mining, data exploration, dictionary development, annotation procedures, and the interpretation of findings.

We have established how language analysis operates as a two-step process: defining the construct and then how that construct manifests in language. The methodological process guided by this two-step framework is seen throughout the history of language analysis in psychology. Gordon Allport formalized the analysis of individuals’ personal writing as a method of psychological research in 1942, reviewing analyses of motivational, mental, and personality characteristics as they manifested through individuals’ personal writing (Allport, 1942). Other psychological constructs considered in early language analysis include delinquency (Healy & Bronner, 1925), friendship (Iovet︠s︡-Tereshchenko, 1936), mental health (Selling, 1932), and unemployment (Zawadski & Lazarsfeld, 1935). Like John Speed, these researchers approached text through content coding, relying on conceptual knowledge to categorize texts as reflecting a given construct. For example, Zawadski & Lazarsfeld (1935) sought to study how unemployment affects people’s psychology, hypothesizing that it would lead to feelings of resentment, anxiety, and suicidal thoughts. The researchers identified these specific constructs by reading 57 autobiographies of unemployed individuals, leading to the evidence-based conclusion that prolonged unemployment causes negative feelings and suicidal thoughts. The particular constructs of interest — negative affect and suicidality — were determined *a priori*. These constructs were expected to be visible in text, such that an attentive reader could answer the questions “is this writer interested in suicide?” or “is this writer experiencing negative affect?”

Drawing psychological inferences from text is analogous to measuring psychological constructs via questionnaires or behavioral tests. If we are interested in measuring a psychological concept — people’s affiliation motivation, for example — we might simply ask participants to introspect and self-report how motivated they felt to connect with others. Presumably, the same motivations also get expressed through behavior. We might record how often people socialize with others, or we might observe whether they choose to socialize when given the opportunity to do so in a lab experiment. In the same vein, people’s motivations may manifest in their language. What kind of language would we expect someone to use when they are feeling in the mood to affiliate with others? Perhaps they would talk about their friends or family or people in general. They might use words relating to social get-togethers such as “party,” “meeting,” “union,” or unifying pronouns such as “we” and “us.” Another possibility is that they express their need for affiliation by describing its absence, using words such as “alone” or “lonely.”

Language analysis relies heavily on the human ability to operationalize psychological constructs as they manifest in language. Either by drawing on one’s own experience of psychological states and used language, or imagining the verbal tendencies of hypothetical individuals, extracting psychological inferences from text data requires placing one’s self within a language user’s perspective. It is a distinctly human ability, much as conversation, or reading a book, or listening to a speech requires the universal human ability to communicate. This, however, presents a pragmatic challenge: how can the subjective, human experience of language be applied towards a framework for objective, evidence-based science? To generate psychological knowledge from language data, a dedicated, systematic approach must be developed.

The Computational Paradigm

A quantitative language science must be built on principles of objectivity, reproducibility, and transparency. This is not so different from other quantitative sciences, but the complexity of language and the inherent subjectivity in assigning psychological meaning to language make the challenge unique. Psychological inferences about language use are rooted in subjectivity, dependent on who is producing the inferences (i.e., the researcher), and difficult to dissect given that they come from inscrutable psychological processes, such as reflecting on the likely intention of the speaker when coding a piece of text. Computation offers a hybridized approach, wherein human subjectivity is treated systematically and inferences are expressed within a statistical framework.

Early language analysis relied on the human ability of coders to understand and interpret human communication, which could sometimes be called into question given the subjectivity involved in language interpretation. Human ratings of texts that unambiguously convey a psychological construct may be objective. But rating ambiguous texts requires subjective judgments. For example, in Zawadski & Lazarsfeld's 1935 study on suicidal ideation, if an autobiography made an explicit reference to suicide, such as "I contemplated killing myself," everyone would code this as suicidal. But the more ambiguous sentence,"I did not see the point of living anymore" would be subject to interpretation. Placing oneself in the shoes of the author (or perhaps their conversant), determining the "truth" of whether there is suicidal ideation is more about mind-reading than objective extraction. Whether a researcher decides to label this as suicidal depends on their intuition and assessment of the internal state of the author and potentially the researcher's own ideas about suicidality and depression. Perhaps the author is describing an existential crisis but never considered literally ending their own life. Or, alternatively, they may simply be speaking in hyperbole. The lack of unambiguous signal is typical in most psychological analyses of text, requiring researchers to identify the pathways of reasoning that led them to their subjective assessment and implement these in their computational approach.

Computation can also offer a more reproducible framework for language analysis than purely subjective coding. Arbitrary factors of the measurement should not influence the findings of a study, though one finds considerable variation in individuals' interpretation and coding of particular texts. One might consider collecting many subjective judgments, devising coding procedures in order to limit the impact of these arbitrary factors. But this solution is a partial solution to a deeper problem, which is that subjective coding is not reproducible from human coder to human coder. For example, in a study of sexist language, male and female coders might fundamentally differ in their interpretations. Female coders might be more sensitive to sexism, understanding the construct at a deeper level, and would potentially rate the same language differently than male coders.

There will always be differences in judgment because there are meaningful differences among people. Comparing text analyses conducted by researchers from different cultures or different time periods introduces inherent differences in judgment (for a detailed discussion, see Krippendorff, 2004). Thus, while manually coding documents within a sound, documented framework is an important component of text analysis, additional considerations must be made in order to meet our standards for reproducible measurement.

The transparency of measurements is an integral component of any scientific field as it permits the identification of error and the observation of new, unexpected phenomena and relationships. However, human judgments about text are difficult to take apart and understand. Why did a coder label a document as suicidal? Why do these two coders disagree? Can we measure constructs in text beyond “I’ll know it when I see it”? While there is no comparison in terms of how computers and humans interpret text — a computer might make errors a human would never, such as assigning a “suicidal ideation” label to a completely non-suicidal text — the principal advantage of the computer in this regard is that we can attribute errors to visible processes, allowing greater levels of interpretation and insight into the constructs we are attempting to measure.

Computerization requires that we make everything in our analysis formal and explicit. Everything — data representation, data processing, algorithms, decisions — is deterministic and programmed. Instructions for a computer in a computational language analysis include text preprocessing (i.e., including and excluding certain characters and words based on predefined rules), pattern matching (finding words of interest in text), and statistical modeling. By scripting the language coding process, we can trace model behavior and gain insight into the measures produced. For example, if we could produce a measure of a text's suicidal ideation by instructing a computer program to follow a sequence of commands, we can justify (and critique) the measure based on the steps that produced it. This is fundamentally different from a measure produced by a human coder, given that human interpretations are not guided by pure formalism but by their own mental processes and insights.

However, a computer's objective advantage over a human is counterbalanced by its disadvantage in the skills and intuitions that humans have for understanding and interpreting language. Humans have evolved as social animals, using language to transmit their internal states with others and navigate social life. Not only can we efficiently use language to convey complex social information about what we are thinking and feeling, we can also use our rich store of life experiences to turn other people's words into meaning. That is why humans are excellent at reading texts and inferring what the author means, what psychological state the author is in, and so on. Computers, on the other hand, have no intuition about the texts they process: they see sentences as character strings and know little about what a text means or how it would be interpreted by a human reader. Thus, computation affords us a variety of advantages, at the cost of losing the intrinsic human socio-cognitive abilities. Below, we discuss a complementary framework for combining the best of computational methods with human interpretation.

Balancing Top–down and Bottom–up Language Analysis

We have discussed the necessarily subjective component of language analysis in psychology — specifically, the insights that a human reader has when interpreting language and the benefits of the computational approach when it comes to objective, reproducible, and transparent measurement. This seemingly presents a paradox wherein neither method is totally satisfactory. In fact, both components can be successfully integrated into a single framework, combining subjective measurement, expert knowledge, and computation. In this framework, top–down methods take expert knowledge and apply them via computational techniques to language data, while bottom–up methods use algorithms from Natural Language Processing (NLP) to represent language.

Top–Down: Offloading expert knowledge

Rather than task computers with understanding psychological constructs such as suicidal ideation, anxiety, or sexist language, experts' knowledge of these constructs must be "offloaded," in some way, to computers. Here, a teacher–student relationship can be used as a metaphor. Given knowledge of a psychological construct, the researcher (the teacher) must provide explicit instructions to the computer (the student) such that the computer can successfully extract the construct from text data. Top–down methods primarily take this approach by developing "dictionaries," categories of words or phrases, each with a distinct link to a given construct or set of constructs. In addition, manual coding can be used in systematic ways (i.e., with well-documented coding guides, theoretical motivation, and measurement of inter-coder agreement) and potentially paired with computational methods in order to automate text labeling.

Bottom–Up: Algorithms applied to language data

The priority of top–down methods like the popular Linguistic Inquiry and Word Count program (LIWC; Tausczik & Pennebaker, 2010), detailed in this chapter, is to extract psychologically relevant and psychometrically valid measures from language. The priority of bottom–up methods, the majority of which have been developed in NLP, is to successfully model language. This is a distinctly different objective from top–down methods; instead, bottom–up methods are guided by research in neighboring fields such as Machine Learning (ML), Artificial Intelligence (AI), and (computational) linguistics. A successful model of language, a concept we will formalize later in this chapter, looks at language as a whole rather than purely in terms of the construct of interest. Thus, bottom–up methods attempt to computationally reproduce many of the functions and aspects of language. For example, probabilistic topic models attempt to automatically cluster, or group, language based on the underlying co-occurrence statistics among words in a corpus (e.g., Blei et al., 2003), statistical approaches to semantics attempt to model the semantic meaning of, and relationships among, words (Mikolov, Sutskever, et al., 2013), and neural language models attempt to model the sequential dependencies of words (Bengio et al., 2003). Towards psychological insight, only a subset of these methods applies, but this subset promises greater accuracy than top–down methods and helps to explore and understand the entire scope of language in relation to psychology.

The optimal combination of top–down and bottom–up methods must consider the psychometric qualities of top–down measures, such as the internal reliability of individual words within a word category, as well as the correctness of a measure’s underlying model of language. With respect to the latter, top–down measures can require assumptions about the underlying distribution of language (depending on whether or not the given construct has to do with the meaning of language, versus just surface-level forms). The top–down approach fails to account for aspects of composition (i.e., words combining in certain contexts to form more complicated meanings), particularly when attempting to measure compositional constructs such as sarcasm. Indeed, this is not necessarily the priority of top–down methods. Contrastingly, the goal of bottom–up approaches is to quantify the patterns that characterize language itself, from the modeling of word semantics using co-occurrence statistics to the clustering of texts based on latent themes in a dataset. Used in a complementary fashion, top–down and bottom–up approaches can yield more linguistically accurate measures of constructs that are “supervised” from the top by theory and psychometric validation. And, as we will show, this combination can be effectively used to explore emergent psychological phenomena in language

Structure of the Chapter

This chapter begins by providing a thorough conceptual grounding in top–down methods, including the development of coding guides and dictionaries; their evaluation and improvement; and their application towards measuring predefined constructs in text data, including word counting software and classification algorithms. The psychology literature provides ample examples of the application of dictionaries, and we survey these studies and highlight the methodological insights that they can offer. Next, we begin our coverage of the bottom–up approach with the “bag of words,” a family of approaches that models word co-occurrence (absent top–down supervision from dictionaries). This includes an in-depth explanation of topic modeling, a key method for exploring and understanding trends in text datasets, as well as building an intuition on how the statistics of word frequency can be used and manipulated to yield fresh insights into psychological constructs.

Following the bag of words, we explain the current paradigm shift in NLP, which pivots to “deep learning” and increasingly complex statistical models. This area of language analysis is complicated and dynamic, with the new methodological standards shifting every few years. We give insight into only a subset of these "deep" techniques, which collectively improve on our ability to explore human language data and to develop models of compositional constructs. Further, understanding these methods is important not only for the practical benefits, but also for adapting to the likely future of text analysis. The new wave of NLP methods is categorically different from many of the previous waves, requiring a new vocabulary, as well as a different toolset. By learning now, the reader is well-prepared to handle the still-evolving set of practices in NLP that can augment, and potentially blend together with, top–down methods.

Lastly, we give the reader an understanding of supervised learning as it relates to text analysis, specifically classification and regression, applied to text data, in order to learn to automatically predict labels associated with the text. The range of methods for this is broad, including conventional statistical techniques such as regularized regression as well as cutting edge techniques from NLP. Similarly, the goals of this analysis are diverse, including prediction (e.g., generating document labels for a large corpus based on a smaller set of coded texts) as well as inference (e.g., determining whether a text and a particular variable are related).

Content Coding Psychological Concepts in Language

Psychologists commonly recruit research assistants to manually code text responses to open-ended survey questions and, as we have discussed, the practice of coding diary entries, news articles, etc., has long been used in psychological research. To generate reproducible and scalable measures from language, these content coding techniques need to be structured according to a computational research design. The primary challenge that this presents is how to represent theoretical constructs in relation to language given the lack of any explicit representation of psychological concepts in machine representations of language. In the top–down approach to achieving representations of psychological concepts in language, there are essentially two strategies: (a) word counting, which involves defining lists of words and phrases indicative of a construct (or of how individuals characterized by a given construct will use language) and using algorithms to find and count these lists, and (b) manual annotation, which involves expert-coding sentences or documents according to a theory-driven typology and subsequent computational analyses using the resulting labels.

Word Counting via Dictionaries

Word counting begins with the insight that psychological states and dispositions are reflected in the words we use. For example, when someone is trying to communicate about their religious faith, they would likely use language related to religious beliefs (e.g., “church,” “pray,” “faith”). This entails an overt connection between words and the psychosocial processes of the speaker. In other words, the words used by an individual provide information about the focus of their attention and their attitudes toward the topic being discussed. Words can also communicate more than the topic discussed. For example, an angry person can indirectly or unintentionally express their emotional state by using language indicating anger, e.g., by swearing an abnormal amount. As evident from this example, in addition to the content that a speaker intends to communicate, the contents of their mind such as the associations, emotions, the focus of their attention, and so on get encoded in our language.

A dictionary enacts hypothesized relationships between communicative intentions and word use, such as those described above. Dictionaries are groups of words that are used to represent a concept (e.g., religion) or which contain markers of psychological construct, state, or trait (Pennebaker et al., 2015; Stone et al., 1966). The positive emotion dictionary might include words like “happy,” “joyful,” and “yay” because these words are usually used to express positive emotions. Similarly, the negative emotion category usually includes words such as “cry,” “sad,” and “angry.” The dictionary approach relies on a core assumption that the degree to which a psychological process or construct is in effect can be inferred from the use of language consistent with said construct or process. Put another way, if words from a particular dictionary category are being used more frequently in a text, it is more likely that the speaker is thinking a certain way, experiencing a particular emotion, and so on. When a dictionary is applied to a text, all the words belonging to the dictionary will be identified and counted. For example, a sentence with more words from the positive emotion dictionary (e.g., “YAY! Today was the best and happiest day”) would be scored higher on positive emotion relative to a sentence with fewer words from the dictionary (e.g., “I had a happy day”), which would be scored higher than a sentence with none (e.g., “Just another day”).

Support for the word count approach comes from a large, long-standing literature demonstrating the predictive validity and utility of tracking word usage. Associations have been identified between word usage and meaningful psychological constructs such as personality (Carey et al., 2015; Pennebaker & King, 1999; Sumner et al., 2011), mental health (Mergenthaler, 1996; Gottschalk & Gleser, 1979; Coppersmith et al., 2015; Pulverman et al., 2017), gender (Newman et al., 2008), political and moral values (Hogenraad, 2003; Alizadeh et al., 2017; Sagi & Dehghani, 2014; Yaden et al., 2018), performance (Robinson et al., 2013), creativity (Martindale, 2007), culture (DeAndrea et al., 2010; Dehghani et al., 2013; Rodríguez-Arauz et al., 2017), and social events (Cohn et al., 2004).

Content Words and Function Words

Historically, text analysis mainly focused on the content conveyed in a text. Consider the sentence “She wanted a cup of coffee.” The sentence has three content words (“wanted,” “cup,” and "coffee") which convey the central meaning of the sentence. Psychologists have developed dictionaries to capture content relating to specific psychological themes such as emotion (i.e., positive and negative emotion), cognitive processes (Pennebaker et al., 2015), moral values (Graham et al., 2009), psychological motivations (Stone et al., 1966), well-being (Ratner et al., 2019), regulatory focus (Kanze et al., 2019), markers of suicidal ideation (Thomas & Duszynski, 1985), and even brand personality (Opoku et al., 2008).

Studies in the last three decades have expanded their focus from linguistic content (what people are talking about) to linguistic style (how people talk). Linguistic style is captured by words pertaining to grammar, including pronouns, prepositions, articles, conjunctions, and auxiliary verbs, which are known as "function words." In the sentence, "She wanted a cup of coffee," words such as "she" and "of" serve the function of connecting the content words in a meaningful way. The significance of function words is highlighted by an incredible statistic: function words comprise less than .05% of the average vocabulary (Baayen et al., 1995), yet roughly 50-60% of the words used in most texts are function words (Rochon et al., 2000; Tausczik & Pennebaker, 2010). The most commonly used words in the English language are "I," "the," "and," "to," and "a" and in Spanish are "de" (of), "la" (the), "que" (that), "el" (the), and "en" (in). Function words can be telling about a speaker's attentional focus and their frame of reference. For instance, in the sentence "she wanted a cup of coffee," the pronoun indicates that the speaker's attention is focused on a woman. It is also clear that the speaker is addressing a listener who shares the speaker's knowledge about who "she" is. Articles mark the introduction of a new object (a cup) or a reference to a known object (the cup). These referential words carry a lot of information about the psychological state of the speaker and the context they are in.

Several studies reveal that much psychological insight is hidden in these traditionally overlooked function words. One of the most studied types of function words are first-person singular pronouns such as “I,” “me,” “my,” etc. which indicate self-focus. Several studies have found that high self-focus, reflected by high usage of first-person singular pronouns, is a marker of depression (Rude et al., 2004) — or negative emotionality more broadly (Tackman et al., 2019) — and susceptibility to suicide (Stirman & Pennebaker, 2001). First-person plural pronouns such as we and us may indicate group or relational identity. For example, football team supporters used high levels of the pronoun “we” after their team’s wins, reflecting how they basked in the glory of their teams’ successes (Cialdini et al., 1976). Other studies have noted that greater article and preposition use is higher in formal language (Jordan et al., 2019) and that such language predicts higher grades among college students (Pennebaker et al., 2014). Students who secured lower grades used more informal language with more auxiliary verbs, pronouns, and negations. Guided by this research, mainstream content analysis programs now include dictionaries for both content and function words. The most popular example of a dictionary-based approach in recent years is the Linguistic Inquiry Word Count (LIWC; Pennebaker et al., 2015; Tausczik & Pennebaker, 2010) which includes dictionaries for content-based categories such as cognitive processes, psychological drives, death and religion, and also function word categories such as pronouns, articles, and adverbs.

Related to the practice of applying expert-defined sets of words for studying psychological constructs in text, studies have also made psychological inferences from the relations between words rather than the words themselves in a top–down manner. WordNet, a comprehensive lexical database of English words, provides a list of the cognitive synonyms associated with any concept and information regarding the links between the various words

(Miller, 1998). Using such semantic networks, it is possible to approximate the semantic distance between any two words or sentences although such distance measures are subject to the accuracy and completeness of WordNet itself. One set of studies measured creativity from the semantic distance of participant-generated ideas from a common prompt initially given to all participants (M. L. Meyer et al., 2019). Ideas that diverged more semantically from the given prompt were scored as more original and creative.

Designing and Applying a Dictionary

One of the appeals of dictionary-driven text analysis is the potential for reuse and reapplication of dictionaries across datasets and studies. The computerized component of LIWC and other dictionary-based programs allows researchers to replicate the same analysis across different datasets by simply reapplying the same dictionary, which facilitates comparisons across studies and between researchers. Here we outline the process of systematically developing dictionaries from scratch.

Formulating Hypotheses about Constructs and Word Use

As mentioned in the introduction, we need to first specify the construct we are interested in and then think about how the construct might manifest in language. In developing scales for psychological constructs, it is common to think about the factors or characteristics of a construct that need to be measured. For example, if a curious researcher wants to build a dictionary that can be used to measure depression, they would hypothesize the characteristics of depression, such as the common cognitive or emotional states that contribute to, or result from, depression, that may also manifest in people's verbal behavior. This step is usually informed by previous research. One of the classical theories on depression posited that depression is marked by a pattern of excessive self-focus and intensified negative affect (Pyszczynski & Greenberg, 1987).

Following from these propositions linking depression with self-focus and negative affect, Rude et al. (2004) sought to identify the language markers of depression. Specifically, they set out to measure two linguistic categories: self-focus and negative emotionality. Where prior work is not helpful, the researcher might rely on their intuition to identify potential linguistic categories that might characterize the construct of interest.

Developing and Refining Word Sets

Once a set of linguistic categories potentially linked to a construct have been identified, the researcher must develop a coding scheme to measure the categories from texts. Without a pre-programmed coding scheme, a computer program would not know how to determine whether the construct of interest is being discussed in a piece of text. Coding schemes can range from the simple word count approach to complex rules involving phrases or strings of words (also known as "n-grams"), full sentences, or even whole documents. Regardless of the type of scheme selected, it needs to be pre-specified to ensure that the analysis is scientific and objective.

The simplest form of the word count approach would involve looking for occurrences of a specific word in the text. If we were interested in the extent to which someone is concerned about their country (e.g., Australia), we might simply count the number of times they use the term "Australia." We might also look for other words related to Australia, such as the names of Australian cities (e.g., Melbourne, Sydney). Similarly, in the most common instantiations of the dictionary approach, we would compute the relative, cumulative frequency of words in the text that represent the target construct. For instance, scoring a text on its expression of anger would require computing the frequency of words in the text which belong to an anger dictionary. To apply this method, we need an anger dictionary in which words relating to anger have been independently identified and grouped together. While dictionaries are readily available for common concepts such as anger, a researcher would need to build a dictionary of their own for less common constructs. How should they identify words representing a concept of interest?

As an illustrative example, we can consider the approach of Dean & Boyd (2020), which used existing knowledge about the markers of depression (e.g., negative affect, self-focus, etc.) in order to inform a computational tool for estimating depression from text. Conceptually, this work began by assuming that a person who tends to engage in high levels of self-focused cognition (a marker of depression) would presumably engage in verbal behaviors such as using language indicating a focus on themselves. To reference the self in spoken or written language, one would have to use first-person singular pronouns such as "I," "me," "my," etc. Texts which use first-person singular pronouns in high frequency (e.g., "I think I am annoyed with myself") necessarily indicate a high self-focus. The researcher might then create a dictionary of all first-person singular pronouns to measure self-focus. Given that there are only a small, finite set of first-person singular pronouns, listing them all is a relatively straightforward task. So is the case with other closed-class function word categories such as articles, conjunctions, and so on.

On the other hand, building a language dictionary for an abstract content category such as negative affect is much more complicated. This is because negative affect is a complex, multifaceted psychological state that may or may not be fully reflected in verbal behavior (Mauss & Robinson, 2009). Even where negative affect is verbally expressed, it can be conveyed using many possible words including "angry," "sad," "annoyed," "darn," "cry," "sob," "dark," "terrible," and so on. If a speaker were posting a message on social media, they might use emojis, exclamation marks, or upper-case letters to convey the intensity of their feelings.

Creative expression (e.g., sarcasm) would render this list endless. How should we identify words that aptly capture the expression of negative affect?

We would need to follow a systematic process of dictionary building. First, a set of candidate words for each category would need to be generated from various sources. We might identify words from questionnaires that have previously been used to measure the focal concept. The developers of LIWC 2007's negative emotion dictionary (Pennebaker et al., 2007) drew on common emotion rating scales, such as the PANAS (Watson et al., 1988). It is also helpful to look at documents that are from known sources or which are known to be making references to the focal concept. To identify words indicating prejudice, we might look in speeches or interviews that have been identified as having racist or prejudiced content. If we wanted to build a dictionary to capture people's political orientation, we might collect the writings of Democrats and Republicans to identify differences in word usage in an exploratory manner. We might also use common tools such as thesauruses, have study participants list words, or just brainstorm with others to generate related words. It helps to collect a number of instances of language use relating to the target concept from several sources in order to identify the many ways in which it can be conveyed and arrive at a rich list of candidate words.

After formulating a list of candidate words for the categories of interest, each word-to-category mapping needs to be evaluated. While doing so, we might identify missed words and include them or exclude words that are rarely used or which are often used to refer to additional meanings, and so on. This process needs to be done in a meticulous manner and can be highly time-consuming. To reduce bias, it is best if more than one person is involved in the process. In the case of LIWC, each category underwent several rounds of evaluation by groups of three judges who independently rated whether each candidate word was appropriate for a particular category. A word was included in, retained in, or excluded from a category only if two of three judges voted to do so (see Pennebaker et al., 2015).

Descriptive and predictive word categories. The choice of words and the sources used in the process would also depend on the purpose or goal of the content analysis. A dictionary can either describe a target construct or be predictive of it. If one were to build a dictionary that describes extraversion, they would gather words that are synonymous with extraversion (e.g., “gregarious,” “social,” and “approachable”). To build a predictive dictionary, they would opt for words that are usually used by extraverted individuals (e.g., “party,” “bar,” and “together”). This distinction is crucial because dictionaries that *describe* psychological constructs such as extraversion or depression may be of little use to psychologists hoping to use them as aids in predicting whether or not the author behind a piece of text can be labeled with the given construct.

Automated dictionary creation. In addition to these largely manual methods, some automated shortcuts can be used. Given a *seed* list (a small set of prototypical words for a given category), researchers have previously used WordNet in order to expand this small seed list into a larger, more comprehensive set of words with similar meanings and senses as the original list (Liu, 2012). In addition to automatically expanding dictionary seed lists in English, which is the most common language used in computational studies of language, researchers have also demonstrated how it can be performed for non-English languages (Maks et al., 2014).

Similarly, latent semantic similarity (rather than manually-defined rules defining similarity among words, i.e., WordNet), can be used to automatically expand a seed list into a comprehensive dictionary (Garten et al., 2018). In this method, two words would be assumed to be semantically similar to one another if they repeatedly occur in similar contexts (Pennington et al., 2014). For example, if we examine a large corpus of language, we would likely find the words “coffee” and “tea” to be repeatedly used in similar contexts (e.g., “drinking a cup of coffee” and “drinking a cup of tea”), and so the two words would be considered to be semantically similar. Using this algorithm, we might compute the semantic distance between the set of candidate words and a large number of other words. Words with the least semantic distance from (or the highest semantic similarity with) the candidate wordlist can be added to the list of candidate words. Increasing the number of relevant words in the dictionary would lead to increased coverage and thereby to more accuracy, but including irrelevant words or words with low base rate (i.e., they are almost never used) would contribute little to the dictionary.[1]

Applying a Dictionary

Frequency weighting and other post-processing. The word counting approach generally involves computing the percentage of words in the text that belong to a dictionary. In this method, all words in a dictionary are given equal weight and are treated as equally capturing a construct. It is also possible to assign different weights to words based on the degree to which they can be thought to capture the target construct. Words that better capture a construct could be assigned higher weights. A framework called VADER (Hutto & Gilbert, 2014) adopts such an approach. VADER’s lexicon comprises words and symbols that were each manually rated for their emotional intensity. Emoticons (e.g., “:D”) and punctuation (e.g., exclamation marks) were also similarly rated. Unlike in the typical approach in which each word is rated on *whether* it is positive or negative, in this approach words were also rated on *how* positive or negative they are. As a result, each word in the VADER lexicon has a sentiment score indicating its intensity of emotional expression. Further, VADER identifies negations (“not happy”) and accordingly

[1] A more in-depth explanation of methods relying on semantic distance is provided in the section entitled “Extracting information from the geometry of an embedding”

inverses the polarity of the sentiment captured. When VADER is applied to a text, sentiment scores are generated at the sentence level.

Validating a Dictionary

A dictionary, like any other measurement tool, must be validated. Validating either a new or an existing dictionary involves determining the definitional, predictive, convergent, and discriminative validity that a given set of words provides. These categories of validity are summarized in Table 1. The definitional validity of a dictionary term set refers to whether the collection of words are 1) coherent around some central concept — e.g., the words “thirsty” and “drink” are closer in meaning than “destroy” and “kangaroo” — and 2) possess face validity with respect to the target categories (i.e., qualitative examination by experts). This can be determined by the researchers or crowd-sourced from a pool of fluent speakers of the language in question.

Table 1.

Types of validity measures for dictionaries.

	Description/Explanation	Examples
Definitional	Whether dictionary items are face valid in correspondence to the construct, and whether items are similar to each other	Do words in a religiosity dictionary have to do with religion or religious concepts? Is each category distinct?
Predictive	Whether the scores generated using the dictionary predict relevant outcomes	Do scores generated from a religiosity dictionary predict religion-related behaviors such as attending religious services? Correlate with self-reported religiosity?
Convergent	Whether the scores generated using the dictionary correlate with other measures of the same construct	Are scores generated from a religiosity dictionary associated with manually coded religiosity?
Discriminant	Whether the scores generated using the dictionary are distinct from measures of dissimilar constructs	Are scores generated from a religiosity dictionary associated with a measure of extraversion?

Predictive validity requires that a negative emotion dictionary should, on average, be correlated with psychological outcomes and behaviors that typically follow negative emotion. In addition, the dictionary's construct validity needs to be tested, which involves two parts: testing the dictionary's convergent and discriminant validity.

The most straightforward method of testing the convergent validity of dictionaries would be to examine whether dictionary-generated scores are associated with manual ratings of a text. This is what Young and Soroka (2012) did to validate the Lexicoder Sentiment Dictionary which was developed to capture the emotional tone of political communications. They also tested whether their dictionary scores correlated with other related lexicons such as LIWC's emotion dictionaries (cf. Pietraszkiewicz et al., 2019). One might also test whether dictionary scores of texts correlate with self-reports of the authors (e.g., Fast & Funder, 2008) or expert ratings (e.g., extracting policy positions from political tests; Laver et al., 2003). Dictionaries can also be validated via experimental manipulations of the construct or other innovative indicators of the construct. For example, Kahn et al. (2007) validated the LIWC emotion dictionaries by using an established experimental paradigm to induce momentary emotional experience and then assessing the use of the emotion dictionary words in participants' oral descriptions of their experience. Collins et al. (2009) showed that words in their mindfulness dictionary occurred at high frequency in a mindfulness manual and also that they co-occurred with the use of active, present-tense words, which is in line with the construct of mindfulness.

The second part of construct validation involves ascertaining that one's dictionary measures only the intended construct and not other unrelated constructs (discriminant validity). We might compute the correlations between dictionary-generated scores with self-reported measures or manual ratings of constructs dissimilar to the focal construct. Pietraszkiewicz et al. (2019) showed that their agency and communion dictionaries were not correlated with LIWC dictionaries that were conceptually unrelated to these constructs. Developers of the mindfulness dictionary mentioned above (Collins et al., 2009) demonstrated discriminant validity by showing little overlap between the mindfulness dictionary and the twelve-step, self-help manual, the "Big Book" of Alcoholics Anonymous, which the authors believed had a conceptual basis that was different from mindfulness. Taking a similar approach, developers of a dictionary of integrative complexity assessed if their dictionary could distinguish philosophical works from the presumably less complex language of lay persons (Conway, Conway, & Houck, 2020). Improving the discriminant validity of a dictionary would increase its accuracy by reducing false positives; for example, how theoretical confounding between expressions of anger and the 9/11 terror attacks can arise from false positives (Pury, 2011). In this case, a dictionary for "anger" mistakenly included thousands of instances of the same message containing the word "critical," while this particular instance of the word was used in the sense of importance rather than emotion. As a result, Back, Küfner, & Egloff (2010) mistakenly inferred that anger amongst Americans sharply increased in the hours following the 9/11 attacks, when this was a result of a dictionary failing to discriminate valid contexts for the target construct.

When a dictionary intends to measure a trait-like individual difference variable, its psychometric properties such as stability and consistency need to be tested (Hussey & Hughes, 2020). That is, if the dictionary is thought to capture a characteristic that is largely stable within individuals over time and across contexts, scores generated using the dictionary should also be similarly stable. Stability can be assessed by measuring the test-retest reliability of dictionary scores at different points of time (Mehl & Pennebaker, 2003) or examining associations between measures from different contexts (e.g., Mehl et al., 2012). Tests of stability are especially important for research on individual differences.

Even well-developed and carefully validated dictionaries rarely capture a concept in all contexts. This is because word usage patterns are not universal. People with different backgrounds (e.g., social class, age groups) or who belong to different cultures (e.g., across regions) may vary in their language usage patterns even when using the same language (see Garten et al., 2018). For instance, the same concept may be described in different ways or the same psychological states may manifest as different linguistic patterns depending on the group or context, which means that a dictionary's ability to capture a construct may vary across groups and contexts. Rouhizadeh et al. (2016) demonstrated how systematic variations in people's use of self-referential pronouns (first-person singular pronouns) mask meaningful psychological patterns. Moreover, as language evolves and new words and phrases begin to appear in popular vocabulary, old dictionaries' become less accurate when applied to texts from these more recent settings. For instance, a dictionary built a decade ago would misclassify or ignore words now popular words such as "TBH," "woke," or "extra." Even a relatively new dictionary may not be equally successful across divergent contexts such as a journal article versus Twitter. That is why dictionaries perform most effectively when applied to contexts that are similar to the ones in which they were tested and validated. For these reasons, pre-built and validated dictionaries too may need to be validated again from time to time especially when being applied to newer contexts. Dictionary manuals should provide detailed descriptions of the contexts that the dictionary was validated in. Another potential solution is to test dictionaries for measurement invariance to determine whether they are equally valid for different groups of people (Hussey & Hughes, 2020).

Manual Content Coding

Dictionaries are applied via pattern matching, which is computationally attractive because the program simply matches words or phrases without recognizing the meaning of language. In order to use dictionaries, we need to be able to identify words corresponding to a construct. However, there may be more complex constructs that cannot be captured by simply looking for synonyms or related words. Identifying these constructs may require the complex social and contextual knowledge that humans use in social interactions. This knowledge can be applied by annotating constructs at the document level.[2]

From the beginning of language analysis in psychology, constructs like suicidal ideation were often coded holistically, beyond the word or phrase. A human's judgment of a piece of text, paired with their knowledge of a construct, allowed them to generate a categorical label. This type of analysis remains an important part of the modern text analysis toolkit: for example, it can provide convergent validity checks on dictionary-based measures (e.g., a human codes a text for its rate of emotional language, which will hopefully converge with the automated measures produced by a dictionary of emotional words). Additionally, it can be used to study constructs that might not be measurable by the presence or non-presence of words and phrases. For example, moral rhetoric can only be partially measured through the presence or non-presence of words like *wrong*, *evil*, or *sacred*. Many communicative activities that humans engage in in language take on this more "compositional" characteristic. It is only with the combination of words in a sentence, in the right arrangement, in a given context, that produces a meaning which is congruent with a prespecified construct like moral rhetoric. For example, the sentence "the current distribution of wealth is fundamentally unfair" draws on socio-political concepts to assert

[2] "Document" refers here to any continuous segment of text, such as a sentence, a paragraph, a social media post, or an entire document. We might alternatively refer to this as "utterance".

a belief about the unfair allotment of resources, whereas the sentence “life is so unfair” is an abstract philosophical reflection that, on its own, does not seem to convey any particular moral valuation.

Designing a Typology and Coding Procedure

Like the development of a dictionary, developing a typology and its associated coding procedure starts with identifying a construct or set of constructs that the researcher believes to manifest in language. For example, Hoover et al. (2020) were interested in applying Moral Foundations Theory (MFT; Graham et al., 2009; Haidt & Joseph, 2004) to language, particularly by using the typology of moral foundations: care, fairness/reciprocity, ingroup/loyalty, authority, and purity concerns. Defining a typology of moral concerns was straight-forward, given the typological definition of MFT itself: each foundation mapped onto a particular class of rhetoric, for example care concerns mapped onto rhetoric that articulated an endorsement or attention to care concerns, or the castigation of violations against care concerns.

Typology development is also influenced by its historical usage in the social sciences, particularly in studying the psychology of political figures and media members through public datasets of their language. For example, political scientists have measured dimensions of personality and cognitive processes (e.g., tendencies to engage simplistic, black-and-white thinking versus multi-dimensional forms of thinking) of U.S. senators, using a systematic content coding procedure (Tetlock, 1981, 1983). Here, the typology used for content coding was influenced first by theoretical foundations (specifically previous hypotheses about the personality profiles of political figures and certain political positions), and secondly by more nuanced textual aspects (e.g., the magnitude or intensity of the construct, directed towards a given entity). Of note for this domain of language analysis, dictionary-based tools have been developed for measuring integrative (versus differential) complexity (see Conway et al., 2020, as well as Chapter 9 in the present volume).

A typology for document-level constructs first differentiates documents of interest from documents of non-interest. In the MFT example above, any document that contains any form of moral rhetoric or the expression of a moral sentiment could be of interest; otherwise, a human coder might ignore the document. Then, for a document that has been coded as containing moral rhetoric, further distinctions would be guided by the respective typology. For example, a human coder could differentiate between “moral purity” and “moral authority” content.

A typology is then applied to natural language data, through the process of annotating texts multiple times by different coders. Multiple steps go into creating a high-quality set of annotations, including having multiple annotators annotate each document and measuring inter-annotator reliability. However, these are safeguards. A well-developed coding procedure, in addition to a sound typology, ensures the quality and reproducibility of annotations, as well as the clear interpretation of the resulting document labels.

Coding guides should include examples of each category in the typology from the domain on which annotations will be performed. For example, if our typology has two high-level categories (“is moral” and “is not moral”), and two second-level categories (“vice” and “virtue”), we ought to gather multiple examples of “moral + vice,” “moral + virtue,” and “non-moral.” These examples should come from the active language domain: for example, if our research goal is to track moral rhetoric on Twitter, our examples should come from this domain. Ideally, examples should be randomly sampled and coded by all involved researchers.

Validating Document Annotations

We have described the benefits of coding a corpus multiple times in order to achieve ratings from multiple coders. This mainly serves as an indication of the clarity of the coding guide and helps to reduce overt subjectivity from influencing our analyses. It also helps to reduce the amount of "noise" in the annotations if, in our final analysis, we take only an aggregation of the set of multiple annotations. If we do this, then outlier annotations (and outlier annotator activity) can be better identified and, if justified, omitted.

The choice of metric for measuring intercoder reliability varies somewhat based on the number of annotators per document and whether the same set of annotators coded all documents in the corpus. In the simple case where two coders assign categorical labels to all documents, Cohen's kappa coefficient (κ) can be used to measure the level of agreement between the two annotators. All intercoder reliability computations use the observed agreement (po) and expected agreement (pe) to express the relative improvement over random chance agreement. Cohen's κ is defined as

$$k = 1 - \frac{1-p_o}{1-p_e}$$

with $p_o = \frac{n_{agree}}{N}$ and $p_e = \frac{1}{N^2}\sum_{i}^{C} n_{1i}n_{2i}$ for all possible categories $i \in C$.

In the case of more than two annotators, the case in which the same two annotators did not annotate the entire document set, or both, Fleiss's kappa coefficient can be used, which is a generalization of Cohen's kappa to an arbitrary number of annotators (Fleiss, 1971), or Krippendorf's alpha coefficient (Krippendorff, 2004). In cases in which the expected agreement is artificially high, which happens for imbalanced datasets with far more negative (i.e., irrelevant) cases than positive, Prevalence-adjusted and Bias-adjusted Kappa (PABAK; Byrt et al., 1993) is one possible solution, which presents a modified observed agreement. Formally, given the number of possible classes C and a previously computed p_o, PABAK is computed as

$\frac{C}{C-1}\left(p_o - \frac{1}{C}\right)$. When C is 2 (the common case in which a label is binary), this simplifies to $2 \cdot p_o - 1$.

Discussion

Content coding is a natural and intuitive language analysis approach for psychologists. The method is deeply rooted in the psychological tradition and shares most of its key assumptions with other psychological methods. The *a priori* specification of theoretically relevant constructs makes the method ideal for testing theory-driven hypotheses. Well-developed dictionaries also fulfill basic psychometric properties such as validity and reliability. Importantly, the popular word count approach is simple to implement and yields easily interpretable measures. For these reasons, the method is accessible and practically useful for psychologists.

One strength of the dictionary approach — its simplicity — is also a limitation in some contexts. If someone uses the word “cry” in a text, a dictionary would likely assume this to be an expression of negative emotion or sadness. It may, however, be the case that the person was referring to happy tears. The author was perhaps referring to a holler (“he cried out”) or the cry of a bird. The author may have even used the word in a sarcastic manner. The word counting approach is blind to the context in which a word is used, which is a huge problem when analyzing short texts such as tweets because a single word would contribute substantially to how a text is interpreted. However, in larger texts, which usually have a large proportion of words relating to the more dominant themes, occasional misclassifications will have little impact on the outcomes. It is also important to note that, across most contemporary psychometric theories in language analysis, the contextualized meaning of a given word is intentionally not considered — use of the word “cry*”* is simply taken as a signal that the author or speaker is attending to a concept (see Boyd & Schwartz, 2021, for a detailed discussion). This nuanced distinction between what a word “means” and what it tells us about a person’s psychology is sometimes difficult, but fundamental (and indeed critical) to an accurate interpretation of most word counting research in Psychology from the 1990’s onward.

Some programs employ a context-aware approach that involves applying pre-programmed context-related rules. For example, VADER accounts for modifiers such as “kinda” and “very” while scoring emotional intensity (Hutto & Gilbert, 2014). Consider these two sentences: “I am kinda annoyed” and “I am very annoyed.” In both sentences, the speaker is perhaps affording comparable levels of attention to the psychological state of annoyance, but the latter sentence is expressing a stronger or more intense experience of emotion. The traditional dictionary approach would assign equal scores on negative affect to these two sentences, but VADER would rate the second sentence as containing more intense negative sentiment than the first one. However, given that contextual influences on language can be complex and that language evolves over time (Louwerse et al., 2004), the costs of introducing such complexity may outweigh the benefits. The complexity of language, which is seen when we attempt to measure higher-order constructs, can instead be addressed from the bottom–up using the statistics of language data and machine learning algorithms. This approach is covered in the remainder of the chapter.

One of the constraints of the top–down approach, which is often ignored in the development and application of dictionaries, is the imbalance in terms of who is developing the dictionaries. For example, prescriptivist attitudes among researchers can bias findings against “non-standard” dialects or sociolinguistic variation,[3] speaking to a deeper issue with dictionary creation, specifically that dictionaries inherently capture the assumptions, biases, and specific

[3] Additionally, most existing dictionaries are in English (cf. Boot et al., 2017; Matsuo et al., 2019)

beliefs about language of the researchers. This power dynamic between the culture of the researcher and the cultures of the studied populations, called the "homefield disadvantage" (Medin et al., 2010) is a roadblock for more equitable and balanced language-based research in psychology. For the reader, knowledge of this roadblock is essential for a thorough understanding of dictionaries and content coding more broadly, and is particularly salient when researchers are working with a non-English language or a population (i.e., language community) different from that of the researcher (e.g., biasing algorithms against language with African American English; Sap et al., 2019).

Text Preprocessing

In the remainder of this chapter, we will outline how meaningful information can be extracted and used to explore phenomena and test hypotheses using "bottom–up" language analysis techniques. Whereas human knowledge can be effectively applied to language data using dictionaries and document annotation, much of the toolbox for language analysis takes a less informed starting point. The data-driven approach, which can also be referred to as bottom–up analysis, includes probabilistic topic models, neural network classifiers, and text embedding. These methods require structured text preprocessing in order to get raw text data into forms that are amenable to statistical modeling.

Text preprocessing roughly follows three steps: cleaning, tokenization, and parsing tokens into "n-grams." In practice, not all three steps are carried out, nor are they necessarily distinct from each other. Multiple programming languages and libraries can be used to carry them out, including the Natural Language Toolkit (NLTK) in Python (Loper & Bird, 2002) and the Text Mining (*tm*) library in R (D. Meyer et al., 2008). Until recently, text preprocessing resources were available mostly for the English language, and the methods and resources reviewed here reflect that bias. Recently, some techniques have become available for text preprocessing in non-English languages, and we note these techniques where appropriate.

Text Cleaning

Recorded language is messy, often containing plenty of characters that are more a product of the medium and less an important object of our analysis. Text cleaning involves qualitative examinations of text files, considerations involving specific media (e.g., social media versus newspapers or diary entries), and applying filters via pattern matching.

Different media types contribute different sets of extra characters and noninformative symbols that should be removed before processing. Newspaper text is carefully proofread, possessing many of the standard features of a language; thus the need for additional cleaning is minimal. Social media language, on the other hand, is dynamic, highly informal, and irregular: words are misspelled more frequently, new words emerge from the dynamism of online social connectivity, and language is casual and often abbreviated. Special symbolic structures, like hashtags (words that start with "#"), hyperlinks, or mentions (words that start with "@") are useful for the particular medium but are potentially not of interest for a particular study. For example, some analyses might only sample Twitter posts using a particular hashtag (thus the hashtag has importance) but, during analysis, the hashtag conveys little additional information. Transcribed speech data present additional issues: when speaking, people have fragmented sentences, start and stop in their speech, and use filler words. Are filler words, such as "um," "er," and "uh" a component of the analysis? What about repetitions, corrections, or false starts?

Another consideration for cleaning text is whether non-alphanumeric characters (i.e., symbols that are not characters or numbers) are of interest. In most cases, across domains, punctuation is removed during preprocessing. While symbols like question marks and ellipses are a natural part of text from a communication standpoint, they are only included in the analysis if our questions directly relate to the usage of those punctuation symbols (e.g., what is the difference in the rate of question mark usage versus exclamation mark usage?). Cases in which punctuation is still valuable to many research questions are apostrophes, denoting possession or contraction, and hyphens that connect compound noun phrases.

Tokenization

“Types” refers to the collection of unique textual forms, and “tokens” are the occurrences of those types in recorded language (Wetzel, 2006). For example, a type might be “fight.” Every time we see the word “fight” in text, this is a token, corresponding to a single type. We could also map the tokens “fought,” “fighting,” “fights,” “fighter,” etc., to the type “fight” each time any one of them occurs in a text (this is known as *stemming* and *lemmatizing* words, discussed below). Alternatively, words can be broken down into separate grammatical components, such as “fight” and “-ing,” which is a more language-general approach to dealing with word variants and different grammatical forms (a popular technique for this is known as *byte pair encoding*, also discussed below). *Tokenization*, in general, is the process of discretizing a string of characters into chunks (usually words but alternatively word subcomponents). Another way of thinking of this process is that a vocabulary (finite set of types) is defined from the ground up by applying a series of processing rules.

Tokenization techniques take strings of cleaned text and parse them into lists of character sequences. Most commonly, whitespace (e.g., spaces, tabs, newline characters) are used to delineate sequence boundaries. In general, text is tokenized for later analysis using a set of processing instructions.

Table 2.

Common Tokenization Implementations in Python and R Programming Languages

	Description	Language/Library/function
Regular Expressions	Pattern matching algorithms that are fast and customizable.	R/tm/Regexp_Tokenizer[4]
CoreNLP Tokenizer	Rule-based, extensive support for many (human) languages.	Python/stanza[5]
Punctuation Tokenizers	Punctuation-sensitive, splits contractions	Python/nltk/word_tokenize[6]
Tokenizers for Social Media	Twitter and other media have specific conventions that need to be parsed by dedicated approaches	R/tokenizers/tokenize_tweets[7]
Model-specific tokenizers	Methods in NLP use tokenizers that automatically identify lemmas, contractions, and some parts of speech	Python/transformers/tokenizer[8]

Tokenization typically involves first determining what constitutes a “word.” In the simplest case, it involves treating all characters as lowercase and uses whitespace (e.g., spaces, tabs, line breaks) as delimiters between tokens. For example, in the sentence “The boy played in the park,” we would have six tokens (“the,” “boy,” “played,” “in,” “the,” and “park”). In less trivial cases, text has meaningful punctuation that factors into tokenization. For example, the string “the book’s title” has “’s” at the end of the book. We can separate “book” from “’s,” leading to two distinct tokens and isolating the analysis of the type “book” from the type “’s” while mapping it to the core “book” type. It might be of interest to consider the pluralization of nouns as its own dynamic, and consider nouns as just their non-possessive/contracted form. This is seen, for

[4] https://rdrr.io/rforge/tm/man/tokenizer.html
[5] https://github.com/stanfordnlp/stanza
[6] https://www.nltk.org/api/nltk.tokenize.html
[7] https://cran.r-project.org/web/packages/tokenizers/vignettes/introduction-to-tokenizers.html
[8] https://github.com/huggingface/tokenizers

example, in the “Punkt” tokenizer in the Natural Language Toolkit (NLTK; Loper & Bird, 2002).

Another example is hyphenated words, for example “well-known.” We must decide whether we think this compound noun is a distinct word, or if we should use the dash to separate the two words into distinct tokens. Later we will talk about n-grams, which consider this type of two-word compound more formally and liberally.

Stemming and Lemmatization

After tokenization, many word forms can be simplified such that they can be mapped to a smaller set of types, by using word stems and lemmas. In English (and other languages, though it is not a universal rule), words have common suffixes, which indicate a different grammatical form, or a different “inflection” of a word. For example, “examples” is the plural of “example,” and “rounder” is the comparative form of the adjective “round.” A word “stem” is referred to in text analysis as merely the base, or stem or root, of a more complex word with grammatical endings, or “inflections,” removed. The process of “stemming” is removing these endings from words in a corpus. A “lemma” is a more linguistically informed version of a stem, such that “fight” is the lemma of “fought.” Both stemming and lemmatization are language-dependent, with stemming being commonly used for English and lemmatization being applicable for any language (though the process of lemmatization is assuredly different between languages).

In many text analysis applications, one finds that stemming or lemmatization is the default practice. However, it is always a step that should be evaluated for the task at hand. The usefulness of stemming and lemmatization are in the simplification of the vocabulary; it is easier to statistically model 5,000 possible types than 8,000, and the difference between “fight,” “fighter,” “fought,” and “fights” is minimal compared to the increased difficulty of accounting (in a statistical sense) for all these more infrequent tokens. Additionally, the size of the given corpus is a consideration for determining whether to perform stemming or lemmatization: with larger corpora, we will likely obtain more reliable (in the statistical sense) frequencies for the various inflected forms (e.g., plurals, tenses) of words. However, the other side to this consideration is that with smaller corpora it is less feasible to keep all the different forms of words. Lastly, stemming and lemmatization might be eschewed when the particular inflections are of interest to a research question. For example, one might want to measure the ways in which people refer to events, themselves, or others with respect to time. Do they refer to events in the past, or in the future, or present? Since this is of interest, we should keep the "-ing," "-ed," and "-s" at the end of verbs. Indeed, the LIWC dictionary contains a number of entries that capture the different inflected forms of words.

Algorithmically, stemming is a pattern-matching approach to stripping suffixes from words, with matching/removal rules specified based on a reasonable knowledge about a language. For English, rules could include removing "-ing" from any word that matches "[xxx]-ing," where the characters in brackets could be any letter. For example, "thinking" is reduced to "think" and "lightning" is reduced to "lightn." In cases like the latter, the limitations of stemming are apparent: for words that match the pattern but are not composed by a "stem" and a suffix, the removal does not make sense. Such errors can potentially be manually corrected through a more complicated ruleset. Popular implementations of stemming include the "Porter" stemmer for English, available through most text analysis libraries, as well as the more sophisticated "Snowball" stemmer for multiple languages in R[9] and Python.[10]

[9] https://www.rdocumentation.org/packages/corpus/versions/0.10.1/topics/stem_snowball
[10] https://www.kite.com/python/docs/nltk.SnowballStemmer

Whereas stemming is based on rules or heuristics, lemmatization is a more advanced strategy that is based on “lemmas,” or word families. A stemming procedure would map “thought” to “thought,” but lemmatization would map “thought” to “think.” Lemmatizers are aware of word structure because they have been manually taught word structure, learned it from training pairs, or given access to a manual mapping between words and their lemmas. Other examples of lemmatizing include reducing “better” to “good,” whereas stemming would not be able to sense what the underlying word family was of “better.” The most commonly-used lemmatizer of this type is implemented for English based on “WordNet” relations among words, and can be applied using the NLTK library in Python.[11]

Lemmatization produces a simpler, more tuned vocabulary for such analysis; however, lemmatizers available to researchers skew towards the English language, as English has been the subject of research in previous decades. Importantly, non-English languages are beginning to be more well-represented in NLP, including the development of lemmatization algorithms, which are built into tokenizers. One example is *sentencepiece*,[12] which uses byte pair encoding (BPE; Gage, 1994) in order to segment words based on the frequency of subword units (i.e., sequences of characters shorter than whitespace-separated word tokens; Sennrich et al., 2015). This data-driven approach provides tokenization that automatically lemmatizes words based on statistics, rather than grammatical rules, and is available for most languages.

Building a Vocabulary

The result of cleaning, tokenization, and potentially stemming is that each text document is a list of tokens. At this point a vocabulary should be established, which is a set of unique types. Several considerations guide this process: (1) how large should the vocabulary be? Rarely

[11] A tutorial for applying WordNet Lemmatizer: https://pythonprogramming.net/lemmatizing-nltk-tutorial/

[12] https://github.com/google/sentencepiece

do we want or need to model every word that appears in the corpus, but neither do we want to make it so small that important words are left out; (2) should we use single tokens as components of the vocabulary, or should we use n-grams to extend the vocabulary? An n-gram is a multi-token type of length *n*, such as “new york” (a “bigram”), “he was tired” (“trigram”), or “one can only hope” (“4-gram”), etc.[13]; and (3) are we interested in extremely high-frequency types (i.e., function words and other “stopwords”) or extremely low-frequency types? Depending on the planned modeling steps, it makes sense to filter stopwords based on string matching (i.e., instruct the program to filter all tokens from a list of “we,” “is,” “to,” etc.) or based on frequency (i.e., filter types from the vocabulary that occur in a large percentage of the corpus).

The goal of pruning the vocabulary is to simplify the modeling tasks while not simplifying too much. The first and third step listed above have to do with pruning the “top” of the distribution (most frequent) and “tail” of the distribution (most rare). Key to both is whether n-grams are used.

The value of constructing an n-gram vocabulary is essentially the benefit of specificity. If an analysis yields the finding that the word “school” is negatively correlated with the age of social media users, it is interesting but hard to interpret. But if the same analysis used bigrams, it might find that the phrases “school sucks” and “love school” are positively and negatively correlated with age, respectively. As a general rule, frequency statistics based on n-grams are more specific, and more *contextualized*, than word-level statistics. But there are restrictions for using n-grams, which are a necessary trade-off of this increased specificity.

One of the most important concepts to understand about n-grams is the trade-off between specificity and the size of a vocabulary. There are, obviously, more unique bigrams than unigrams in any corpus. One can visualize this in Figure 1, where a large corpus of public

[13] Though not discussed in detail in this chapter, an n-gram of length 1 is a “unigram”

domain text from the Gutenberg project, web text from user reviews, and the Brown corpus (Kučera & Francis, 1970), have been cleaned and tokenized to show the distributions in frequency among unigrams, bigrams, and trigrams. The most frequent unigram appeared 100,000 times, while the most common trigram appeared only 1,000 times.

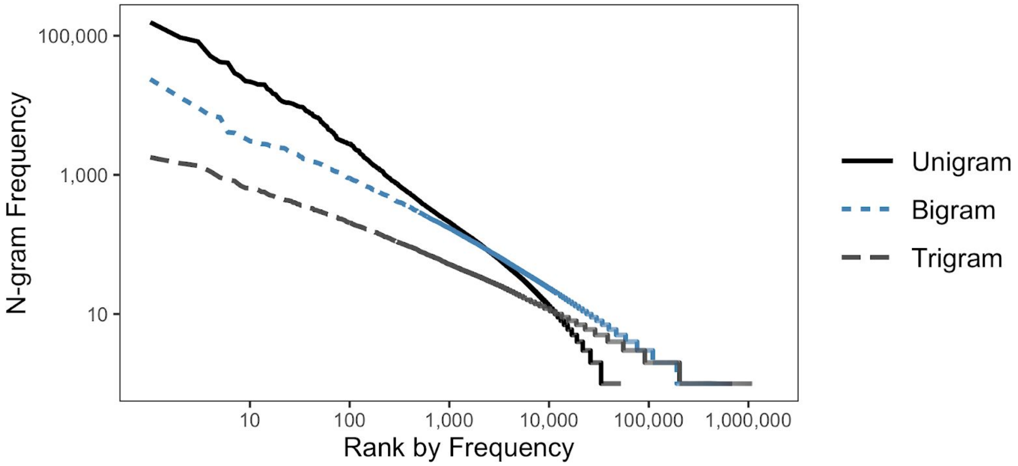

Figure 1. N-gram frequencies plotted by highest frequency rank (on a log-log scale) for a corpus of English books, web forum comments, and news articles. A straight line indicates conformity to the Zipfian distribution, where the rank of an n-gram is inversely proportional to its frequency.

The distributions of unigrams, bigrams, and trigrams pictured in the figure follow the Zipf distribution, and are thus called “Zipfian.” In other words, the rank of a word in a corpus (by frequency) is inversely proportional to its normalized frequency (Powers, 1998). So, if the word “the” constitutes 6.9% of the tokens in the corpus, and is the most frequent word, then the second-most frequent word should occur half as often (we find in this corpus that “and” has a frequency of 4.2%). All n-gram sets follow the Zipfian distribution, and as n gets larger, the slope of the line in Figure 2 decreases in magnitude. This means that the tail is getting longer, and the number of rare, highly-specific types in the vocabulary is increasing. These higher n-gram sets are essentially an extension of the tail of the vocabulary, as each unigram can be combined to form exponentially more bigrams, each bigram can be combined to form exponentially more trigrams, etc., and each of these exponential extensions make the tail longer and sparser.

For small corpora, using n-grams does not make sense, as anything more than a unigram will be mostly sparse (i.e., most n-grams do not appear in the majority of documents). But with larger text corpora, getting more specific than unigrams is less problematic. A popular use-case of n-grams is the Google N-grams corpus, which provides the normalized frequencies of millions of n-grams in a massive corpus of English over the past century (Michel et al., 2011). The Google N-grams project is an excellent example of the benefits of n-grams, as unigram trends across time are sometimes less informative than bigram, trigram, and other n-grams in tracing culture and specific types of messages, rhetoric, and events. For example, the trend lines via the Google N-gram Viewer[14] for the unigrams "iron" and "curtain" are relatively unchanging over time; however, the bigram "iron curtain," referencing an historical event, peaks sharply in the 1960s.

Whether a corpus or modeling task requires unigrams or n-grams, it is generally desirable to filter the resulting vocabulary. There were over 50,000 unique unigrams in the corpus described by Figure 1, and more than 1,000,000 unique trigrams. A conventional cut-off used in text analysis is the first 10,000 types by frequency are included, the rest pruned from the vocabulary. Recent research has specifically investigated this conventional number, finding that a vocabulary size of less than 10,000 (e.g., 5,000) can achieve the same predictive performance in text classification as 10,000, using a formal statistical approach rather than a heuristic (Chen et al., 2019). Therefore, using a fixed, absolute threshold for vocabulary size is likely too arbitrary. Instead, one can prune the vocabulary with rules (e.g., pruning infrequent and overly frequent types) and leave the overall size not subject to a preset limit. For example, the most frequent

[14] https://books.google.com/ngrams

types in the vocabulary are also pruned, as these are mostly "stopwords." In subsequent sections we will revisit why stopwords are often removed for statistical analysis, but the basic reasoning is that their extreme frequency in proportion to the rest of the corpus (e.g., stopwords are many times more frequent than other words) biases models away from content words. Practically speaking, stopword lists are provided in most text analysis libraries to facilitate easily pruning them by directly matching strings — for example, the NLTK library in Python provides stopword lists in multiple languages[15]. Lastly, a qualitative examination of the vocabulary, in particular the most frequent terms, can indicate whether undesirable types are present in the vocabulary, such as catch phrases, hashtags, or other artifacts of the medium or particular dataset. These can then be filtered directly by string matching.

Text Preprocessing Outcomes

The goal of text preprocessing is to set the stage for modeling. In the next section, we will discuss the "bag of words" approach which, as the name implies, considers documents as sets of tokens without considering order. This approach starts with a cleaned and tokenized corpus, potentially stemmed or lemmatized, a vocabulary of n-grams (possibly just unigrams) filtered, and a matrix of counts, where the rows are instances/documents and the columns are types or n-grams in the vocabulary, and the contents are counts of those types in each instance. This is called the "Document-Term Matrix," or Doc-Term Matrix for short.

Lastly, many applications using neural networks, which are covered in the section after bag of words approaches, merely represent each instance/document as a sequence of word/type identifiers, where each identifier corresponds to a type in the vocabulary. These are then fed into neural networks to generate vector representations, initialize text classifiers, or other sequence-sensitive models. In general, these techniques do not use n-grams, merely a unigram

[15] Chapter 4, section 1 in the NLTK book: https://www.nltk.org/book/ch02.html

vocabulary that maps words (or parts of words) to unique identifiers, since sequence and co-occurrence information is captured in other ways in the modeling step.

Word Frequency from the Ground Up

In the dictionary paradigm, prior information is represented by word groupings representing distinct psychological concepts, such as positive emotion or social language, or categories of function words, such as first-person pronouns or past tense. This provides a direct link between theoretical constructs and natural language data. But there are limitations with assuming so much about word meaning. Dictionaries are fairly specific, rigid "priors," which presuppose quite a bit about language. Rather than extract variables from natural language that fit our prior conception of a given construct, unsupervised machine learning can be applied to natural language data without necessarily assuming a fixed set of word meanings with dictionaries.

This section covers practical methods for deriving data-driven representations of language. Starting from the "bag of words" and Latent Semantic Analysis, practical and conceptual building blocks for unsupervised learning from text, we cover progressively more advanced techniques, in terms of statistical machinery and fitting procedures.

The Bag of Words

Unlike the dictionary-based word counting technique, which is a top–down heuristic, here we introduce the first in a series of bottom–up methods seeking to parse language into interpretable quantities for analysis. The first idea introduced by the "bag of words" (as opposed to *a priori* dictionaries) is that word categories, and indeed the meaning of words as they occur in text, are defined by the patterns with which these words are used. If a dictionary defines a word family with the word "mother" in it (among others), the sentences "I love my mother" and

“You’re an ugly mother f*****” would produce the same output for the category containing “mother.” We might then say that the top–down approach is “naive,” as it ignores the behavior of words in documents and heavy-handedly maps categories onto documents. Now, a computational method that perfectly represents these multiple meanings of “mother,” and of course the meanings of other words that have different usages and functions across contexts, is hard to formulate. This is evident throughout this chapter, as we describe more and more complicated methods to represent the varied meanings of words and sentences. The first step towards such methods, however, begins by conceptualizing language through the lens of words, the co-occurrence of words, and the statistics that these more primitive variables produce.

The “distributional hypothesis” proposes that “You shall know a word by the company it keeps!” (Firth, 1957). This idea is the basis of data-driven language representation, versus top–down applications of prior information. Statistical learning algorithms can learn their own representation of which words ought to be grouped together, and which are functionally equivalent or similar in usage. The distributional hypothesis specifically states that we should make use of the fact that two words are probably similar in meaning if they are used in similar ways. From our example earlier, over a large corpus of language one might come across contexts in which “mother” is used in an insulting manner (i.e., with the word “ugly”), and one in which it is used in a loving manner. The multiple senses of “mother” are discoverable from statistical patterns in usage rather than a mental view of what the word means, because the usage contexts change between the two senses.

Bag of words modeling, which is motivated by the distributional hypothesis and applies this thinking by counting and analyzing sets (i.e., “bags”) of words, takes the document term matrix, discussed in the previous section on text preprocessing, as a starting point, and attempts to statistically extract useful (i.e., predictive of non-linguistic phenomena) and interesting measures from text.

Term Frequency–Inverse Document Frequency

A common transformation to apply to the document term matrix, which has been shown to greatly amplify its usefulness and interpretability, has to do with boosting the weight of rare terms relative to highly frequent terms. This is called "Term Frequency-Inverse Document Frequency" (TF-IDF). For example, function words like "the," "a," "is," and "was" appear so frequently in text that deriving meaningful co-occurrence statistics with these words is basically useless (i.e., they occur in virtually every document). Much more relevant are words that have higher variance (e.g., content words), the presence of which can tell us a lot about the content of a document. This reasoning is counter to the thinking in dictionaries, wherein the relative frequencies of function words can be meaningful because of the information that they are *a priori* known (or believed) to convey. But that reflects the differing goals of dictionary and bag of words analysis: here, we are trying to maximize information from word occurrence statistics, whereas dictionaries do not expressly capitalize on within-corpus co-occurrence patterns.

Let us suppose that we want to explore the variation in style and substance of news articles. What are the linguistic factors that are most impactful in driving such differences between articles and domains? We start by looking through a set of words and how their occurrences vary across documents, for example, "town," "the," and "filibuster" (these words are arbitrary and for illustrative purposes only). "Town" is not necessarily a common word in English like "the," but it is relatively common in a corpus of news covering political interests. But "filibuster" is infrequent enough that only a few articles — those discussing the use of the political technique of the filibuster — will have the word. This is precisely the type of information we want to amplify. We want to tell our models that this signal is useful, and we want to increase the weight of such words in our models enough that any occurrence of them would give us a great deal of discriminatory power between the speaker and the others in the dataset. The relationship between frequent and infrequent words in a corpus — occurrences of the frequent ones tell us comparatively little, whereas infrequent words' occurrences can tell us what makes a text "unique" or special — drives us to pursue mathematical techniques to amplify the signal from infrequent words and dampen the signal from frequent ones. This technique, developed decades ago and still the best for its purpose, is called Inverse Document Frequency weighting (IDF; Jones, 1972).

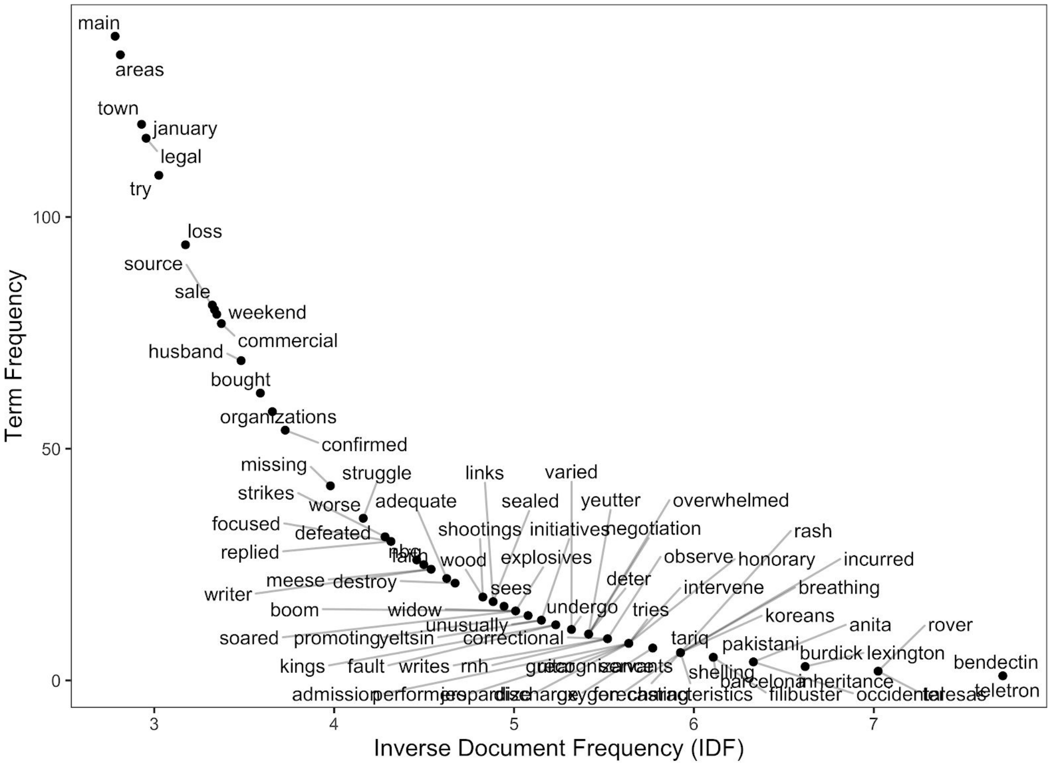

Figure 2. The frequency of words plotted against their IDF weight, for a random subsample of the Associated Press corpus via the *topicmodels* package in R. High frequency words have a low IDF, and the majority of words have lower frequency and higher weight.

The definition of the IDF weighting schematic varies across textbooks, papers, and software implementations, but the most common is defined in Robertson (2004):

$$idf(t_i) = log\frac{N}{n_i}$$

A term, t_i, is given an *idf* weight of the logarithm of the ratio between the size of the corpus (N) and the number of documents in which t_i occurs (n_i). The distribution of total counts and IDFs for a corpus of news articles, including the words referenced above, is shown in Figure 2. Note that IDF is computed by document counts ("is the word in a document?") and is insensitive to a word occurring multiple times in the same article, whereas the count displayed is the overall frequency. In the figure, stopwords like "the" have been filtered during preprocessing because we already know they do not interest us from the bag of words perspective. The count n_i of "town" is high, and correspondingly *idf*("town") is low, around 3. But for rarer terms like "filibuster" the number of occurrences is low thus *idf*("filibuster") will be higher.

IDFs are computed for each word (or type) in the vocabulary, and then for every occurrence of the word in the document term matrix, the original weight is multiplied by the IDF. Conventionally, in order to minimize the influence of the length of particular documents, term frequencies are also divided by the length of the document (known as the "term frequency" normalization).

The TF-IDF normalization step was, and remains, an extreme improvement on document-term matrix representations. In information retrieval (IR), which is generally concerned with retrieving relevant documents/text entries based on a query, TF-IDF vectors have been one of the longest-lasting innovations, and they are still capable of providing reasonable performance in text classification and text matching.

N-gram Language Modeling

We introduced n-grams above in the section on text preprocessing. One of the most basic applications of n-grams is to compute conditional probabilities, thereby constructing what is called a "language model." An n-gram language model is typically based on the learning objective given by estimating the probability of observing a word conditional on having observed a given (n-1)-gram immediately before. For example, a trigram learning objective we would estimate (by extracting these statistics from a corpus) the probability of seeing the word "tired" given that we have observed the bigram ("he," "was"). These probabilities can be obtained by first preprocessing and tokenizing a text corpus and then counting every n-gram and, for all possible words that follow each n-gram, compute the frequency and normalize by the n-gram's frequency. Using these probabilities, an n-gram language model is used to give the probability of observing a document (i.e., a sequence of tokens), based on the aggregation of all the conditional probabilities given by the n-gram objective. For more details on n-gram language models, we recommend more authoritative sources, such as Chapter 3 in Martin & Jurafsky (2009), and implementations of n-gram language models, such as the Natural Language Toolkit (NLTK[16]; Loper & Bird, 2002).

In practice, n-gram language models are not a practical tool anymore because they have been superseded by neural network language models. They are important to comprehend, however, because the concept of language modeling is fundamental. One of the main ways that NLP researchers have estimated the distribution of language is by deconstructing language into this "next-word" prediction task. N-gram language models are the most intuitive approach to this, using simple corpus frequency statistics to estimate the likelihood of the next word given an observation of the previous words.

Latent Semantic Analysis

[16] https://www.nltk.org/api/nltk.lm.html

This next technique focuses on compressing the information of a document-term matrix, such that rows (documents) and columns (words) can be expressed in fewer dimensions. Why is this compression desirable? One reason was articulated by Deerwester et al., (1990), and relied on the idea that words took on different meanings when occurring in different contexts. *Polysemy* points to the fact "that most words have more than one distinct meaning," for example, the English word "bank" can be a river bank or a financial institution. *Synonymy* points to the fact that "there are many ways to refer to the same object" (p. 392), for example, "happy" and "joyful" are interchangeable. Practically speaking, this refers to two types of errors we could make if we relied only on the count of a word in a given document: false positives, when we incorrectly specify that a word is a part of a category when it is being used in a different way (polysemy) or miss that a word is indeed part of a category, because it is being used in an identical way to a word within the category (synonymy). The secret to algorithmically detecting polysemy and synonymy lies in effectively recognizing patterns in word co-occurrences throughout the document term matrix.

Latent Semantic Analysis (LSA) is an application of a powerful method in linear algebra called Singular Value Decomposition (SVD) and is part of a larger family of methods in machine learning for "dimensionality reduction." SVD approximates the most compact representation of the observed data that preserves the maximum information. If given a TF-IDF matrix of size 1000x5000 (representing 1000 documents described by 5000 words/n-gram features), SVD/LSA reduces 5000 to some number of dimensions K, where K is selected beforehand as the desired size of the reduced matrix. Three key outputs come out of an LSA program: a new matrix of size 1000xK, representing the reduced data; a matrix of size Kx5000, giving the loading of each word onto each dimension; and the ratio of total variance explained, per dimension. The explained variance is particularly important, as it is both an indicator of the success of the algorithm and can be used for “tuning” K, i.e., selecting the K which explains a sufficient amount of the total variance of the matrix, without being redundant (having dimensions that do not explain much variance). The reduced matrix (1000xK) is useful for prediction and document-level statistical analysis, while the word matrix (Kx5000) is used as a “low-dimensional” representation of words, a precursor to “word embeddings,” which are described in detail in a later section.

The reduced matrix gives parsimonious document representations useful for predictions (e.g., Garcia & Sikström, 2014), though we will describe more novel methods that are often better for this purpose, and interpretable word representations, which both can be used for visualization, clustering, and computations of similarity. The initial motivation behind LSA was to provide document indexing (retrieval based on a query), which explains the other popular name for LSA, Latent Semantic Indexing (LSI). This indexing allowed users of information management systems to query documents using combinations of query terms, and not just single terms, since they could be compared (i.e., their similarity could be calculated) based on the low dimensional representation of documents and the query-document. Thus the pseudo-document example consisting of the terms “mother” and “love” could be projected into the low dimensional space, and documents that were most similar to the resulting projection would be returned, as opposed to all documents with just “mother.” This is a trivial example, but serves to illustrate the power of using dimensionality reduction methods based on matrix algebra for representing words and documents. Additionally, LSA illustrates some of the basic structure of data-driven text analysis: compact representations of words and documents based on observation.

In small corpora, LSA is a strong, quick first step in generating word- and document-representations from DocTerm matrices. Its mathematical elegance, based on widely used principles of matrix algebra, lead to a variety of desirable properties, including the ability to quantify documents' similarity to dictionaries (Sagi & Dehghani, 2014). The most important idea that LSA generated is that observed language can be explained by a combination of latent semantic categories, which are essentially just groups of words within distinct semantic subdomains. In a sense, this is part of what an *a priori* approach to language analysis (i.e., dictionaries) attempts to define, with the important difference being that groups of words co-occurring together are not necessarily the particular groups of words defined by a dictionary.

Probabilistic Bag of Words Models

Although LSA remains useful as a technique for explorations of small corpora, its main historical contribution has been to motivate probabilistic methods for achieving the same outcomes as LSA, but with greater validity, scalability, and functionality. The core idea is that language representations can be learned through statistical models, rather than count-based measures and matrix decomposition. Probabilistic Latent Semantic Analysis (PLSA) was proposed to bring a statistical modeling perspective to LSA (Hofmann, 2001). Specifically, in contrast to LSA, PLSA defines a "proper generative data model," meaning that "... standard techniques from statistics can be applied for model fitting, model selection and complexity control. … More specifically, PLSA associates a latent context variable with each word occurrence, which explicitly accounts for polysemy" (p. 178). This insight was key to actually capturing word polysemy, a task at which LSA struggled. More important was the conceptual transition from the "count and compress" method of LSA, towards probabilistic data modeling.

For practicality's sake, we will only discuss the most effective and popular general probabilistic model for the bag of words, and in addition several variants that expanded on this model. Latent Dirichlet Allocation (Blei et al., 2003) is a latent variable model that assumes that words in documents are generated by sampling from a mixture of “topics,” where each topic is a cluster of coherent, commonly co-occurring words. Technically, a topic is a distribution of the words in the vocabulary, with probability values ranging from 0 to 1, and with words included in the topic having higher values. The power of this “generative” model is that documents can be sampled from the chain of distributions defined by the model — the mixture of topics, each topic’s cluster of words, the sampling of documents from selected topics — and that we can subsequently update the parameters of the model by computing the closeness of the generated documents to the actual documents.

Before fitting an LDA model to data, it resembles a “recipe” for a topic model. It has no actual content yet, only the parameters and structure that can describe any arbitrary topic model. Much like a Bayesian statistical model will update its probabilities based on its “priors” and new data, a topic model will update its probabilities based on its priors (default distributions for words belonging to topics, and for documents being generated for a certain selection of topics) and new data (observations of words occurring together in documents). An LDA model that has been fit to data is thus best interpreted according to both its initial “recipe” as well as the information it learns from a dataset.

Structurally, a topic model resembles a latent semantic model, or even an *a priori* dictionary to some extent. It defines clusters of words that are derived exclusively from data (versus expert knowledge), but with greater predictive validity than LSA. Topic models can be used in analysis in much the same ways as LSA: examining the relatedness of documents (in the space of LSA dimensions or LDA topics), visualizing trends across corpora, analyzing document covariates in regression models, or, by using LDA variants, incorporating additional information into the model fitting process, such as target keywords that focus attention on certain semantic domains, metadata, or hierarchical structure.

Implementing a Topic Model with LDA

Models like LSA or linear regression have a closed form solution that allow the algebraic computation of optimal parameters (i.e., the parameters and weights of the model that are the best fit to the observed data). LDA, like other probabilistic algorithms, have much more complicated structure, requiring either variational inference (Blei et al., 2003) or collapsed Gibbs sampling (Griffiths & Steyvers, 2004). Briefly, variational methods approximate the difficult likelihood function described by the topic model by solving another, easier likelihood function, while sampling methods iteratively get closer to the true "best" model by sampling. In practice, variational inference is faster than Gibbs sampling, deterministic (it will produce the same topics over multiple runs if you use the same "random seed"), and is less "accurate" with respect to the true topic distributions (Geigle, 2016; Teh et al., 2007). A Gibbs sampling implementation of LDA is slower to train, but will theoretically converge at the optimal topic assignment with infinite training iterations. In practice, researchers have found implementations of Gibbs sampling to be superior in the resulting set of topics.[17] Accessible implementations of Gibbs sampling are provided by Mallet,[18] a Java-based library, the python package "lda,"[19] and the "lda" package in R.[20] Variational methods, which are faster to train and can produce comparable results to Gibbs sampling, are implemented by the python package Gensim[21] (Řehůřek & Sojka, 2011) and others.

[17] See (Resnik et al., 2010) for a primer on the implementation of a Gibbs sampler for LDA, and Goldberg, Y [@yoavgo] (2020) for a discussion on the advantages of sampling methods versus variational (approximation) approaches

[18] http://mallet.cs.umass.edu/index.php

[19] https://github.com/lda-project/lda

[20] https://cran.r-project.org/web/packages/lda/

[21] https://radimrehurek.com/gensim/

Another consideration in beginning a topic model implementation is choosing the right topic model for the corpus. Topic modeling has been used widely for modeling news articles with clear topical themes, blogs, websites, academic papers, etc. In psychology, our corpora often have different characteristics from these regular forms. They might include tweets, video captions, dialogue turns, or transcriptions from spoken language. This kind of language has less grammatical consistency, less length (i.e., information to use during model fitting), and can be domain specific. Thus a number of considerations must be taken into account. First, LDA has been shown to be ineffective at modeling small texts like Twitter posts (Hong & Davison, 2010), intuitively explained by the fact that short documents have insufficient information for the typical LDA generative story. Instead, other LDA variants have been proposed for short texts, most notably BiTerm Topic Modeling (Cheng et al., 2014; Yan et al., 2013), or LDA can be used in clever ways in order to pool short texts into longer, informationally rich documents (Mehrotra et al., 2013).

Another consideration has to do with text preprocessing. As discussed above in the context of more general n-gram co-occurrences, high-frequency words render topic models uninformative. If words like “the” and “is” are included while fitting an LDA model, the best fit to the data will have topics dominated by these words, since explaining their high-frequency occurrences drowns out the signal from less frequent words that we still want the model to be able to explain. This is roughly the same logic for applying TF-IDF transformations to word frequencies. Another preprocessing consideration is stemming. In topic modeling, stemming is often performed in order to shrink the vocabulary, making the task of finding an optimal set of parameters more tractable. Additionally, the definition of a “topic” by most people would be redundant if “love,” “lovely,” “loved,” “loves,” and “loving” are all in the same topic. However, there are some circumstances in which these words are not, in fact, redundant, but the norm for topic modeling is to perform stemming or lemmatization in advance of model fitting.

The main set of considerations for topic modeling are the selection of "hyperparameters." Like all parametric learning algorithms, LDA has hyperparameters, or meta parameters, which are chosen *a priori* and control meaningful properties of the produced model. In LDA, the most important hyperparameters are the number of topics (known as the *k* parameter), the *alpha* hyperparameter, and the *beta* hyperparameter. The latter two affect the distributions of topics within documents and words within topics, respectively. A lower *alpha* will lead to documents being assumed to have a few, dominant topics, while a higher *alpha* leads to documents being assumed to have many, non-dominant topics. A lower *beta* will lead to topics with more weight on a few, dominant words (i.e., the probability mass will be focused on a narrow set of terms), while a higher *beta* leads to topics that more evenly allocate their probability among words. These parameters are largely guided by one's expectations of the data (are documents short and topically focused, or long and topically diverse?) and one's desired properties of the topics (do we want topics to be highly distinct, or more inclusive to a greater portion of the vocabulary?). In most studies, the default values for *alpha* and *beta* are used. The most common cases in which one would alter these default parameters is when a particular type of topic is desired, or if it is believed that a corpus contains documents better modeled by higher or lower values of *alpha* and *beta*.

The last key parameter to consider is the number of topics, or *k*. Again, there is no objectively correct answer, and small variation in *k* (e.g., from 25 to 26) are often inconsequential. Selecting *k* can be guided by the user's knowledge of the size of the corpus (i.e., one would not select *k* of 25 for a corpus of 10 million, or *k* 1000 for a corpus of 1000), or potentially its complexity (i.e., higher *k* for more diverse vocabularies). For example, an arbitrarily large *k* (e.g., 2000) can be chosen in order to "over-capture" signal in a very large corpus of millions of documents from tens of thousands of individuals, and subsequently topics can be pruned based on correlation with exogenous variables (H. A. Schwartz et al., 2013). On the other hand, a small corpus (e.g., 3,500 documents) might require *k* closer to 20 or 30. In most cases, the number of topics ought to be subject to some experimentation. Some variations on LDA attempt to automatically derive the optimal number of topics, for example the Hierarchical Dirichlet Process (Teh et al., 2005).

There seems to be an arbitrary nature to topic modeling that some may find unsatisfactory, both aesthetically as well as for the validity of any psychological inferences that come from it. The arbitrariness is real, and can only ever be partially mitigated through validation measures which we will discuss below. But it bears emphasizing that LDA should not be the sole tool used in text analysis, and its outputs should never be unconditionally trusted, particularly if the model has not been evaluated in other ways (e.g., hold-out data, "close reading of topic-labelled texts), or if the model has only been fit once without rerunning to ensure stability.

The important outputs of an LDA model include the topic models (distributions over vocabulary, i.e., sets of probability weights per word), and the topic likelihoods per topic, per document. Thus a corpus can be described by each document's "loading" onto *k* topics, just as the output of an LSA produces documents loaded onto *k* dimensions, or factors.

Evaluating and Interpreting Topic Models

A generative topic model is difficult to evaluate and interpret, as its outputs do not come from closed-form equations (LSA, i.e., SVD) or *a priori* word categories. On the one hand, there is a requirement that some subjective evaluation is used to judge a topic model, since thus far we have managed to completely offload the “thinking” component of the analysis to probabilistic inference. On the other, there are procedures and metrics researchers have proposed to make the evaluation and interpretation of topic models more objective.

Topic models are a useful tool for exploring trends in a corpus of documents, and produce more effective bag of words features than LSA. They are also trusted in practice. Just as interpreting human language requires inferences about intentionality and context, interpreting topics is a fairly subjective practice that can be made objective only by combining human evaluation with rigorous investigation of the model’s fit to the data. Hidden underneath colorful visualizations and exciting performance can lie a mess of overfitting, bad preprocessing, and a lack of transparent interpretation.

One of the most viable techniques for evaluating a topic model is to let data — specifically, non-textual data such as psychological or demographic measures — tell us which topics are important, or “externally predictive.” This can be called “extrinsic evaluation,” which relies on outside sources of information to validate the reliability and correctness of a topic model. Above, we noted the approach of choosing a large number of topics and letting the data speak for themselves (H. A. Schwartz et al., 2013). This is carried out by fitting regularized regression models, using the topics as predictors, to predict questionnaire responses or demographic variables. Thus the predictive topics can be selected as the most meaningful, and the burden of interpretation can sit on the assumption that the topics signal valuable information.

But one may not have access to such large stores of non-textual, psychologically-relevant data. One way that topic models are evaluated and interpreted is through qualitative means. The first such method is the most time consuming, and involves collecting human (non-expert) judgements about the topics in order to judge their internal coherence. Several approaches have been proposed for this, most notably the two tasks proposed by Chang et al., (2009). The *word intrusion task* gives testers six randomly ordered words, five of which are from a given topic. If the five words are coherent, then the intruder is easily identified. This gives a per-topic measure of a model. The *topic intrusion task* measures the model's overall ability to describe the corpus. Given a document title, a snippet of the document, and four topics (represented by the top 8 words in the topic), the tester must identify which of the four topics was low probability for the document, versus the three high probability topics for the document. These two manual checks can be expensive, but when the quality of a model is prioritized, or when comparing two alternative topic modeling approaches, it can be a necessary component of the research.

The less systematic qualitative measure is just a glance by a human over the topics. The coherence of the topics produced by a topic model can be assessed at face value based on the apparent coherence from each topic. If the top three words of a topic are "mother," "day," and "love," the natural coherence of Mother's Day might spring to mind, and we can roughly judge that this particular topic is picking up on a coherent theme. But if another's words are "bastard," "table," and "lighthouse," perhaps the topic is picking up on a select few documents about bastards sitting at a table in a lighthouse, and that this is not so much a general trend through the corpus as it is an idiosyncratic artifact of the algorithm.

Finally, there are some quantitative methods for determining the "fit" of a topic modeling. These are based entirely on the data themselves and require no human intervention. By dividing a corpus into "test" and "train," where the train corpus is used to fit the model and the test corpus is used to validate the model, one gets an empirical estimate of a given model's generalizable fit to data. Typically, the validation measure is the negative log likelihood the model assigns to the test data. If the negative log likelihood is low, then the topic model has overfit to the training corpus or is a bad fit in general to the data, but if it is high, then the topic model is a good fit for the corpus in general. This is a relevant practice especially for choosing the parameters of a topic model.

The usage of LDA topics in word clouds and other visualizations can create the appearance of coherence and clarity that is not always there. Chang et al. (2009) notes the tendency of both psychologists and computer scientists to abstain from “looking under the hood” of topic models, trusting the assumptions of the powerful LDA model and the software used to fit models to data. Though several metrics have been suggested to measure the overall fit of a topic model, such as log likelihood, coherence scores (Mimno et al., 2011), and the manual “intrusion” tasks outlined above, researchers must accept a degree of uncertainty, reporting metrics as well as how the effects in a given analysis change with alterations in the hyperparameters, training approach, LDA variant used, and the particular random seed used.

Topic Modeling Variants

LDA is merely the starting point for a set of methods that can be used for modeling language in the bag of words setting. Indeed, the LDA model itself is often referred to as “Vanilla LDA,” given that no modifications have been made. Depending on the requirements of a given corpus and analysis objective, different modifications can be made. We will highlight two LDA variants — structural topic models and seeded LDA — and provide pointers to others.

Topic modeling, in general, is about answering psychological questions about text. Two questions that topic model variants can answer in pursuit of this goal are (1) can topics be related to external measures, such as demographic variables, psychological outcomes, or annotated labels, and (2) how do probabilistic topics relate to existing, prespecified word categories? The former refers to a set of methods which attempt to directly supervise topic models during training (i.e., resulting topic distributions over words are simultaneously predictive of external covariates and coherent in the traditional LDA way), versus supervision after an LDA model is trained and documents can be represented with topic likelihoods. The Structural Topic Model (STM) uses prior distributions in order to coerce documents with similar metadata (e.g., speaker political party, location, or the time of occurrence) to have a similar topic distribution and distribution over topic co-occurrence. For example, if we were to compare a vanilla LDA model with an STM, where metadata consisted of a binary gender variable, the STM topics would more often describe themes that are common in language by one of the gender values than the LDA topics. In other words, STM topics might miss clusters of co-occurring words that are randomly distributed across metadata, but provide more precise examinations of the language patterns most important for studying particular metadata variables.

The latter refers to models which introduce “lexical priors” (Jagarlamudi et al., 2012) into topics, or “seeds,” such that certain topics are constrained to include prespecified words but are free to come to include other words during training. This is essentially a way to “nudge” topics to target certain classes of language (e.g., a dictionary category like positive emotion words) without exclusively modeling the words in that category.

Other variants of LDA have more to do with achieving a good model fit, as opposed to achieving some additional objective. The BiTerm Topic Model (BTM; Yan et al., 2013) was proposed for the specific purpose of modeling short documents. By modeling term-term co-occurrence rather than documents as mixtures of topics, the BTM gets around a fundamental issue with LDA, which is that it performs poorly when observations (i.e., documents) are short, as is the case with Twitter posts and open response questions in surveys. The Contextualized

Topic Model (CTM; Bianchi, Terragni, & Hovey, 2020) is a novel method for combining the advantages of neural language models (covered in the next section) with LDA.

Table 3.

Variants of LDA

Variant Type	Description	Implementations
Vanilla LDA	Dirichlet allocation model with collapsed Gibbs sampling or Variational EM	R/topicmodels, java/mallet, R/lda, python/lda, python/gensim
Seeded LDA	Topics are initialized with seed words, or “lexical priors”, to encourage topics to be more coherent or to target specific domains	Anchored Corex (python)[22]; R/topicmodels
Structural Topic Model (STM)	Includes outcome variables (e.g., metadata) at document level. Documents with similar metadata will have (a) similar topics, and (b) similar combination of topics.	R/stm[23]
BiTerm Topic Model	Specialized model for short documents (e.g., Twitter posts).	R[24]
Contextualized Topic Model (CTM)	Pre-trained neural language models are used to improve topic coherence using text embedding. See next subsection for details.	Python[25]

The Open Vocabulary Approach

One of the principal motivations for the bag of words is that dictionaries, since they are created *a priori* based on experts’ knowledge and subjective processes of refinement, are liable to miss important linguistic patterns in corpora relative to outcomes and variables of interest. In other words, it is not guaranteed that predefined dictionaries capture coherent aspects of language usage. Whereas a dictionary approach might specify the types of words used by those high in Extraversion, a bottom–up approach would let the data speak for themselves, finding

[22] https://github.com/gregversteeg/corex_topic
[23] https://cran.r-project.org/web/packages/stm/stm.pdf
[24] https://cran.r-project.org/web/packages/BTM/index.html
[25] https://github.com/MilaNLProc/contextualized-topic-models

clusters of words that are correlated with individual-level variables. Bag of words representations, in particular LDA, are a useful tool for discovering new categories of words by correlating them with individual-level variables and outcomes. In the "Open Vocabulary" approach to language analysis (H. A. Schwartz et al., 2013), patterns in language are discovered in a bottom-up manner that are correlated with dimensions of personality (Park et al., 2015), symptoms of depression (Guntuku et al., 2017) and markers of schizophrenia (Mitchell et al., 2015), and geographic distribution of variables such as "well-being" (H. A. Schwartz et al., 2013).

One of the essential methodological innovations of these works is the collection of social media data from participants. One strategy is to have participants volunteer their personal data (i.e., demographics) and participate in validated surveys and questionnaires. Several works (e.g., H. A. Schwartz et al., 2013) and other works have made use of the My Personality project (Kosinski et al., 2013), which allowed users to take personality surveys and volunteer their Facebook accounts for research. Other works studying mental health, both in the ability to detect disorders and to determine the linguistic correlates of various conditions, have used Twitter users, depending on users posting publicly about their diagnoses and symptoms (Coppersmith et al., 2014, 2015; Mitchell et al., 2015).

Studies that pair individual-level measures of psychological traits, demographics, and mental health diagnoses with social media posts often rely on a combination of top–down (i.e., "closed vocabulary") and bottom–up (i.e., "open vocabulary") methods. Predefined dictionaries are grounded in theory and typically undergo extensive internal and external validation. Thus, the measures they produce in this context are readily interpretable. However, top–down methods inherently make strong assumptions about the language data they are being applied to. There might be categories of words that are relevant to a particular construct (e.g., personality) or a particular domain (e.g., Facebook) that are not captured by dictionaries. It is reasonable to complement dictionaries with topic models, such as LDA. Topic models are not always easily interpretable, as we have discussed; however, through validation through internal measures (consistency, coherence), visualization of topics, and comparison of various topic modeling choices such as structural topic models or seeded models, they can provide bottom–up insights that are unavailable to dictionaries.

One recent example of combining bottom–up and top–down methods to explore and understand an existing psychological construct is Kennedy, Atari, Mostafazadeh Davani, Hoover, et al. (2020), which studied the relationship between moral concerns and language via Facebook status updates. Domain-specific dictionaries were used to answer targeted questions, specifically whether a given moral concern (e.g., "Fairness") is related to using moral language of the corresponding category (e.g., "Fairness" language). LDA, among other models, was then used to explore patterns of language usage that were distinctly related to each moral concern.

Supervised Modeling with the Bag of Words

A common application for bottom-up techniques, in general, and bag of words representations, specifically, is supervised modeling. Supervised modeling for text analysis is the practice of fitting a model to predict a label or outcome from text-based inputs.

There are possibly two different objectives of supervised modeling on text. First, sometimes we might have labeled documents (e.g., tweets labeled for the type and strength of sentiment expressed), and we want to build a text classifier that performs maximally on held out samples, so that we can apply that classifier to a much larger, unlabeled corpus. Second, we might attempt to infer the relationship between text and label, especially when that label carries meaning in terms of a construct (e.g., a manually coded label from a theoretically-informed coding typology) or conveys information relevant to a hypothesis or target phenomena (e.g., the political party of the author of a text). Here, we will introduce the first, more established set of methods for supervised modeling, primarily based on the bag of words. In the following section on deep learning techniques, we will revisit supervised modeling in the context of more recent methods that achieve superior predictive performance than bag of words features.

Bag of words methods yield informative continuous variables that can be used in traditional statistical models, such as regression, or more novel (though still established) machine learning methods like the Support Vector Machines. In other words, we have an input feature matrix ***X***, which has observations in rows and features in columns, and a corresponding set of labels ***Y***. We will first discuss the different types of ***Y***s, what goes into constructing ***X***, and what methods are available for this type of predictive modeling.

Types of Modeling Tasks

Above, we identified two settings in which supervised modeling is appropriate. First, human-generated labels can be used as ***Y***, indicating the sentiment/affective dimension of a tweet, the rating of a text for its expression of suicidal ideation, or whether a text is an example of a target construct, such as moral rhetoric. This type of modeling is clearly informed by the discussion of expert-coding above. In this setting, prediction (as opposed to inference) is typically the primary goal. Second, metadata which are to be associated with text (e.g., the political party or gender of the speaker/author, the time or day of utterance, the number of likes on a social media post, etc.) can also be modeled as ***Y***. Here, *inference* (as opposed to prediction) is typically of greater interest. Bag of words methods are not necessarily optimal in terms of predictive performance (as we will see, deep learning methods are typically more effective in terms of generating accurate, out-of-sample predictions) but are straight-forward in terms of inferring relationships among text and label.

The feature matrix ***X***, which in the case of the bag of words approach can consist of the document term matrix (with or without TF-IDF normalization), the reduced matrix from LSA, the probabilities from an LDA model (indicating the probability, for each document or observation, of each topic in the model), or a combination of the above. Dictionary-based measures can of course be used in supervised models, wherein each category outcome (i.e., word count normalized by document size) is a single input variable, and it can make sense to combine dictionary variables with bag of words variables in supervised learning (see below).

Predicting labels for new samples. One approach involving text classification is training predictive models on labeled corpora and subsequently using these models to predict the labels of out-of-sample data. This is a setting in which the ability to generate valid predictions is prioritized. For example, in the setting of analyzing the language of mental health disorders, the priority is often to be able to detect when an individual is exhibiting symptoms based on their language. For detecting individuals' exhibiting symptoms of schizophrenia on Twitter, Mitchell et al., (2015) trained a variety of text classifiers using features based on dictionaries, LDA, n-grams, and all these features combined, finding that a model using LIWC features and LDA features was the best-performing classifier.

Inferring relationships. The second major setting in which supervised modeling is called for is when analyzing the relationship between text and labels in a dataset. Most often, these labels are of theoretical interest, and might be used to test hypotheses. There is more importance, therefore, on the feature set, rather than on the method or optimizing the method's performance. Choosing a feature set is guided by the goals of the analysis. Targeted analyses, which test hypotheses or compare explanations of a phenomenon, will benefit from top-down, parsimonious feature representations. For example, one might use a small set of dictionaries on affect in order to test a theory about individual differences in emotion. Less targeted analyses, which might take a more exploratory, or "existential" approach to testing a theory or attempting to find a certain relationship, involve a more "kitchen sink" attitude towards feature selection. Often, TF-IDF vectors, topic model probabilities, or similar features can be used with predictive models (with careful model validation) to (a) measure the existence and magnitude of relationships between labels and text, and (b) to identify important features that drive the discovered relationship. Many open vocabulary methods use this approach, identifying new relationships based on sparse regressions or supervised models.

In either setting, measuring model fit via cross-validation or similar methods is essential. This is because of the possibility of over-fitting, since text data tend to be high-dimensional and the models we use for supervision can pick up on spurious signals.

Methods for Feature-Based Supervision

A number of machine learning techniques can be used for learning classifiers or performing regression given the feature matrix $\boldsymbol{X}$ and the set of target labels $\boldsymbol{Y}$. These methods balance prediction (being able to find patterns that generalize well to out-of-sample data) with interpretability. Importantly, each is practically viable: a variety of programming languages and statistical resources provide easy access to each. The priorities of the given modeling task inform the choice of model to use, and in many cases several, if not all, models can be used in order to inform comparisons. Lastly, especially in the purely predictive setting, these feature-based methods are best used as baselines for more sophisticated models. For example, Kennedy, Atari, Mostafazadeh Davani, Yeh, et al., (2020) compared a feature-based model (using TF-IDF vectors) to a leading method in NLP for predicting whether social media posts contained hate-based rhetoric.

Table 4 contains a brief summary of the most important methods for supervised modeling for text with features. Most often, these methods are applied with bag of words features (e.g., TF-IDF, LDA probabilities, etc.). Each can flexibly be used for text classification (predicting a categorical label) or regression, and the most effective models for predictive use regularization in order to reduce the odds of over-fitting to high-dimensional text data.

Table 4.

Three Notable Feature-Based Prediction Methods for Text Data

	Description	Pros	Cons
Regression (Hoerl & Kennard, 1970; Tibshirani, 1996)	Regularized regressions are a solid baseline for modeling text data	Allows multi-class classification and feature interpretation	Weaker in performance than SVM
Support Vector Machine (Corinna Cortes & Vapnik, 1995)	"Splits" data points of a class by finding a separating plane; can model non-linear relationships[26]	High performance with bag of words features (Joachims, 1998)	Expensive with large *N*; no direct probabilistic interpretation
Naive Bayes (Rennie et al., 2003; Sang-Bum Kim et al., 2006)	Estimates the probability of each class for each feature	Clear interpretation of feature importance and outcome probabilities	Poor performance with correlated features

Validating and Interpreting Supervised Models

Great care must be taken when performing text classification or regression not to overfit, and to report statistics on out-of-sample data. Especially in the case of classification, reporting metrics need to be carefully selected and understood, as measures like model accuracy are taken at face value without considering the base rates of target classes.

[26] Non-linear SVMs are supported by the "kernel trick", which efficiently computes similarities among data points in higher dimensions (e.g., quadratic or exponential transforms as opposed to linear coefficients)

Performance metrics for text classification. Most classification metrics are only informative relative to baselines, or heuristics and simpler models. For example, one of the target classes might be 1%; a “naive” classifier, which always “guesses” the majority (99%) class, will score an accuracy of 99%, which is completely uninformative yet impressive-looking. More detailed metrics, such as *precision*, *recall*, and *f-score* (also known as “F_1”), give a much better indication of the performance of a classifier. Briefly, for a binary classification task, *precision* computes the ratio of correct predictions of a given class relative to the number of times that class was predicted — alternatively, the number of true positives (TP) divided by the sum of TP and the number of false positives (FP). If a classifier has low precision, then it is predicting the positive label far more often than it should, and has a high false positive rate. *Recall*, on the other hand, computes the ratio of correct predictions to how many true positives there were — alternatively, TP divided by the sum of TP and the number of false negatives (FN). If a classifier has low recall, then many of the true positives are undetected.

In some classification settings we prefer high *precision*, as we want to minimize false positives; in other cases, we might prefer high *recall*, as we do not want to risk missing any positives. More commonly in classification, we want to balance these two. The *f-score* statistic computes the harmonic mean of precision and recall:

$$\frac{2 * precision * recall}{precision + recall}$$

f-score will not only balance precision and recall, but also penalize more relative imbalance. For example, *f-score* will be lower for *precision* and *recall* of 0.33 and 0.67 (respectively) than for *precision* and *recall* of 0.4 and 0.6, even though the average of both pairs is the same.

The default setting for *precision*, *recall*, and *f-score* is binary classification, when one of the two classes is clearly positive and the other negative. For example, a dataset of tweets with some labeled positive for “moral rhetoric” and others with no moral rhetoric. When multiple classes are present, these metrics are aggregated in one of two ways: macro averaging computes each class’s scores and then averages them, treating each class essentially equally; micro averaging computes *precision*, *recall*, and *f-score* based on the global counts of FP, TP, FN, and TN. Micro averaging is preferred in the setting where certain classes are much more or less frequent than the others. In the binary classification case, this decision does not need to be made.

Out-of-sample validation. As text data become more high-dimensional, and the predictive model becomes more complex, in-sample performance becomes more meaningless. Moreover, interpreting such models (e.g., examining models’ coefficients corresponding to individual features) is highly unreliable if a model has only been evaluated on the same data on which it was trained. This is due to the likelihood of overfitting. Briefly, overfitting is a phenomenon wherein a statistical model fits the training data *too* well, capturing patterns in the data that do not generalize outside the training dataset. For example, a model is overfit if it achieves 90% *f-score* on a classification in its training data, yet only 40% *f-score* when evaluated on a dataset of the same population and label set. Thus, cross-validation performance for classification and regression models should be reported, or at the minimum performance on a single held-out validation set. And additionally, model coefficients should only be interpreted for models that have been validated.

Discussion

The bag of words is a cornerstone of language analysis, due to its simple elegance, accessible interpretations, and tractable statistical modeling. As we will see in the following section, once we move towards modeling sequences and the compositionality of language, the modeling task becomes exponentially more difficult. Approaching the bag of words in applied research requires balancing its trade-offs with both dictionary modeling and more sophisticated bottom–up techniques.

The bag of words was coined thusly: “[F]or language is not merely a bag-of-words but a tool with particular properties which have been fashioned in the course of its use” (Harris, 1954, p. 156). Indeed, it is easy to see that representing language as unordered sets of term frequencies departs significantly from language as it is used by humans. Imagine that, hypothetically, one were to communicate solely with the bag of words; hardly anything would get done. But still, one must admit that this is the goal of bottom–up modeling: we want our models to be an accurate reflection of reality, to actually capture the observed phenomena. It was this reasoning in the first place that motivated statistically modeling the bag of words rather than merely counting, and that motivated bottom–up word counting rather than *a priori* word categories. The same reasoning will motivate the use of neural networks to model language.

But there is no magic formula by which the same algorithm can always be applied, no model that can be applied in all circumstances. In machine learning, the “No Free Lunch Theorem” states that “... if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems” (Wolpert & Macready, 1997, p. 69). For text analysis, no single approach is sufficient for every analysis. Dictionaries are a powerful tool for measuring well-articulated constructs in language, but are less able to capture patterns in a corpus that are not specifically captured by dictionary categories or word frequency. Bag of words techniques and the open vocabulary are tools that are often best-suited for exploration and, in cases, prediction, but they are inherently disassociated from theoretically guided measurement tools and can thus be less useful for hypothesis testing and theory comparison.

The No Free Lunch Theorem for text analysis reflects its multi-dimensional goals. *A priori* constructs must be quantified, the results of modeling must be interpretable and yield qualitative insights, and prediction must be in cases optimized. The bag of words is a foundation of text analysis because of its role in fulfilling some of these objectives, and will likely remain as part of the text analysis toolbox even as bottom–up methods become more sophisticated.

Deep Learning for Text Analysis

In the past decade, deep learning methods have become the standard for modeling language data. Deep learning, which is a broad term to describe multi-layer networks of artificial "neurons" (LeCun et al., 2015), refers to a family of data-driven techniques that can approximate any function (Hornik et al., 1989) yet are generally viewed as "black boxes," which produce predictions from inputs without providing insight into their own predictions.[27] Nevertheless, they are the leading paradigm in most data intensive fields, including health informatics (Ravi et al., 2017), computer vision (Voulodimos et al., 2018), time series modeling (Ismail Fawaz et al., 2019), and natural language processing (T. Young et al., 2018). Deep learning is a model-building framework for prediction and for generating unsupervised feature representations — for example, numeric representations of words based only on unlabeled text corpora. Its role for analytic tasks, such as understanding psychological constructs in text or exploring trends in corpora, is very much in flux: computational researchers continue to develop more interpretable techniques and strategies, such that the future of deep learning in text analysis is bright, if not yet fully realized. In this section, we explain the key concepts of deep learning as it relates to language, the best practices for language analysis, and the current trajectory of the field.

[27] As we note in the general discussion of this chapter, research into interpreting and explaining "black box" neural network models is an active area of research. Methods have been proposed which offer interpretations from neural networks, though these are not yet established practices.

Deep learning methods are an extension of the “bottom–up” philosophy typified by models like LDA. Neural networks[28] offer a similar approach to, for example, LDA, only with more modeling capacity. In NLP, this capacity includes the ability to optimize the prediction of linguistic modeling objectives, such as predicting the “next word” in a sequence or predicting the distribution of a word’s “context window” (see Table 6). With the deep learning approach to language, we can greatly improve our predictions (of external variables, such as text metadata and linguistic structure, such as parts of speech) and representations of language.

Organization of this Section

We begin this section by introducing several important concepts which may be unfamiliar to the reader. First, we discuss the notion of a “good” representation of language, and how this motivates our use of advanced techniques and motivates still more innovation and application in language analysis. Next, we further motivate deep learning for language analysis by highlighting the “curse of dimensionality,” which explains why language in particular is so difficult to model in a bottom–up manner, and why neural networks are suited to the task. Finally, before beginning our main coverage of deep learning techniques, we articulate the key constraints facing those using neural networks in applied settings, and the tools at our disposal for handling these constraints. In particular, we discuss the difficulties of training neural networks, the problems presented by their lack of interpretability, and the tools of “transfer learning” and algorithmic interpretation.

Two main families of techniques are covered in this section, which are in fact similar in structure. First, *word embedding* involves learning to map words into a continuous vector space, similarly to the way that LSA can provide latent vectors of words based on a document-term

[28] We will use “neural network” and “deep learning” interchangeably. Technically, deep learning refers to networks that are “deep” in the sense of possessing multiple (i.e., 4 or more) layers (Schmidhuber, 2015), though networks in NLP often do not meet this requirement.

matrix. We provide some historical context, highlighting the key methodological innovations as well as summarizing word embedding variants and how to apply them. Second, *text encoding* refers to the process of modeling sequences of words in continuous vector space, with the goal of capturing the compositional and grammatical aspects of language. In particular, *pretrained language models* (e.g., Devlin et al., 2018) are a leading, breakthrough method that we will introduce. Again, we will spend time developing an intuition into the successes of these encoders, as well as provide some standard practices which have emerged for their use in applied settings. We also note that, in comparison to earlier sections, this section is the most technical and includes the most mathematical detail.

An Overview of Deep Learning for Language

Simply put, neural networks are able to learn "good" representations of language. Producing good representations of data is the fundamental goal of "representation learning" (Bengio et al., 2013), which has seen rapid progress with the rising prominence of deep learning. Though a representation for something does not necessarily have to be purely numeric, as is the case with word embeddings, for example, and can in fact be relational or symbolic, we will consider representations, and the qualities which make representations "good," through the lens of numeric representations. In a mathematical sense, Bengio et al. (2013) proposes that a good representation is "... one that captures the posterior distribution of the underlying explanatory factors for the observed input," and "one that is useful as input to a supervised predictor" (p. 1798). In simpler terms, we might say that the quality of these representations is assessed from their correctness and usefulness, respectively. These two criteria are interwoven, such that it is difficult for a representation to be useful without explaining the underlying explanatory factors of observed data (and vice versa). For example, deep representations of language are the leading method for applications such as translation, answering questions, or engaging in dialogue with a human (Goldberg, 2016). Hence, some researchers conclude that deep representations capture the structure of language (i.e., the syntax and semantics of languages) implicitly (Tenney et al., 2019).

Deep learning methods complement LDA and other probabilistic bag of words methods, extending the same modeling paradigm in key ways. The extension that neural networks provide to LDA and similar methods is their ability to fit complex distributions, of which language is one of the most complex. Language data are unavoidably high-dimensional, creating an incredibly difficult challenge from the perspective of statistical learning. The "curse of dimensionality" for language can be described thusly:

> A fundamental problem that makes language modeling and other learning problems difficult is the curse of dimensionality. It is particularly obvious in the case when one wants to model the joint distribution between many discrete random variables (such as words in a sentence, or discrete attributes in a data-mining task). For example, if one wants to model the joint distribution of 10 consecutive words in a natural language with a vocabulary V of size 100,000, there are potentially $100000^{10} - 1 = 10^{50} - 1$ free parameters (Bengio et al., 2003, p. 1137).

In other words, there are a lot of words a human could possibly use and virtually infinite ways in which these words might be combined together. This points merely to the fact that there are nearly infinite functions for language, depending on the communicative goals of those involved and the particular contexts of the language in question. Given this, a statistical model will always struggle to model language, simply from the perspective that there are so many events to handle. With n-grams, for example, increasing from unigram, to bigram, trigram, etc., makes it harder and harder to estimate quantities from data, as the number of unique n-grams increases exponentially. Neural networks offer a partial solution to the curse of dimensionality.

The Mechanics of Neural Networks

Neural networks (a) are arbitrarily good function approximators, and (b) scale with the training data. In other words, neural networks are flexible learners that thrive when given access to copious amounts of examples on which to tune their parameters. With large enough text datasets, neural networks can approximate the distribution of words (in sequence) far better than a method that models word frequency and co-occurrence. The process by which this occurs, and the necessary components to achieve this modeling outcome, are outlined below.

There are generally two sides to fitting neural networks to data. First, one must select a neural architecture and a training strategy or loss function. An architecture specifies the configuration of artificial "neurons," or perceptrons (Rosenblatt, 1958), into modeling layers. Architectures are customized for certain types of modeling problems; some architectures are more predisposed to capture sequential relationships, while others are better suited to model images. The most common and general architecture is the feedforward network, or multi-layer perceptron (MLP). An example of a simple feedforward network in NLP is the *skip-gram* architecture, used to learn word embeddings (Mikolov, Sutskever, et al., 2013). The Recurrent Neural Network (RNN) is an architecture proposed to model sequential data, or specifically processes which evolve over time (Elman, 1990), such as words in a sentence. The RNN architecture[29] has several variants, particularly the Long Short-Term Memory network (LSTM; Hochreiter & Schmidhuber, 1997) and the bidirectional RNN or bidirectional LSTM (Graves & Schmidhuber, 2005). The Convolutional Neural Network (CNN) is a popular architecture for image processing and similar applications, and has been successfully used for language processing tasks, such as text classification (Conneau, Schwenk, et al., 2017).

These are the building blocks of learning representations of language from data. In many cases, more complicated networks are constructed using these more basic building blocks; for

[29] Andrej Karpathy's blog provides an introduction to RNNs: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

example, later we will introduce the Transformer architecture, which is composed of feed-forward networks. Perhaps more important than understanding networks, however, is understanding modeling objectives, or loss functions. Any statistical model has some objective, such as minimizing the residual sum of squares (for a regression) or maximizing accuracy (for a classification). In terms of model fitting, the weights of neural architectures are tuned with "backpropagation"[30] (LeCun et al., 1988; Rumelhart et al., 1986). The representations a model learns is a reflection of the loss function chosen: LDA and other bag of words methods learn to model the distribution of word frequencies in a document; word embedding methods learn to predict the distribution of words in small context windows; and text encoding techniques learn the likelihood of sentences given surrounding sentences, and the likelihood of masked words in a sentence (Devlin et al., 2018). When applying a neural network method, the objective function tells a story about what is learned.

The other side to training neural networks is the actual model fitting, which first involves understanding one's dataset, and matching it to the right modeling strategy. It is a well-known fact that deep learning requires large datasets in order to work. Thus the size of the data available for training is a major consideration. If the dataset of interest is small (i.e., less than several thousand observations), then training a neural network only on this data is likely unfeasible (we will discuss below the option of "transfer learning," or of reusing components of previously trained networks). In addition, the type of the dataset, specifically whether the dataset has labels, informs whether unsupervised or supervised learning is employed. In the first setting, neural networks' capacities can be maximized by learning representations from unlabeled text, which is called "unsupervised learning." For example, *skip-thought* vectors (Kiros et al., 2015) were

[30] An introductory blog post on backpropagation: https://missinglink.ai/guides/neural-network-concepts/backpropagation-neural-networks-process-examples-code-minus-math/

trained using a large collection of books, using only the ordered sentences in each. If instead we have labels for a dataset and we want to learn to perform “supervised learning” (learning to predict the labels of a dataset), neural networks can be used in combination with embeddings, discussed later, or pretrained language models.

Lastly, training a network is a process that requires trial-and-error, implementation heuristics, and sound engineering practices. These cannot be fully covered in this introductory chapter; however, many learning resources[31] for deep learning can offer instruction for understanding network regularization (e.g., “dropout”), learning rates, batch training, optimization, and techniques like “early stopping.”

But, fortunately, many of these implementation practices are only marginally useful within applied settings, as it is often unnecessary to train a network from scratch. One of the most important developments in NLP is the rise of “transfer learning,” which has made it possible for applied researchers to conduct neural network-assisted text analysis without the burden of training entire models.

The Trajectory of Deep Learning for NLP: Transfer learning and beyond

The idea of transfer learning is that machines should reuse, or “transfer,” the knowledge learned from one prediction task to the next. For example, pretrained word embeddings can be downloaded[32] instead of generated each time an applied researcher wants to analyze a corpus. In recent years, transfer learning has become the most rapidly developing approach in NLP (Ruder et al., 2019), and the future of language analysis is most likely tied to this ability to reuse and analyze previously-learned representations of language (similarly to the manner in which human children do not learn language “from scratch”).

[31] One recommendation is the following set of tutorials: https://machinelearningmastery.com/category/deep-learning/

[32] For example, “GloVe” vectors can be downloaded in plain text form (https://nlp.stanford.edu/projects/glove/)

Another way to understand transfer learning for NLP is by thinking of language learning in the human context. Given that a human requires years of cognitive development and nurturing, on top of millions of years of evolution, in order to become competent in language, we can think of computers' difficulties in quantifying language as suffering from a short memory. If we "start from scratch" every time we analyze a language dataset, we are unrealistically asking computers to do what we could not: make sense of a huge set of symbols, taking into account many layers of cultural and psychological meanings as well as the complexities of syntax and semantics. Thus, rather than starting from scratch (as we do in LSA or most variants of LDA), we can inject our analyses with developed representations of language.

This development has huge implications for psychologists and others wanting to analyze language data. For example, in text classification, transferred representations can be plugged in to the classifier, effortlessly boosting performance by leveraging information learned previously from large text corpora. As a result, datasets do not need to be prohibitively large in order to use neural network techniques.

Word Embeddings

Word embeddings refer to a broad class of methodologies that have become a staple of NLP research in the past 10 years. Interestingly, the first word embeddings were based (and continue to be motivated by) the idea of distributed word semantics, pursued via methods such as LSA. The application of the distributional hypothesis, in particular, continues to be the foundation of representation learning for language, extracting semantic information about words from their "neighbors," or the other words with which they frequently co-occur. As we will show, the effectiveness of neural networks for modeling co-occurrences is in the ability of models to capture local relationships among words, as well as global features of words' co-occurrences over large text corpora.

As PLSA and LDA evidenced, prediction is a more successful representation learning paradigm than counting (Baroni et al., 2014). Indeed, representations that were learned by fitting a predictive model (versus performing dimensionality reduction on word count statistics) were objectively "better," using our above notion of a good representation. Along the same lines, the initial word embedding models used prediction in order to model the meaning of words. Below, we describe in detail the workings of these innovating models, and how they can be used for generating predictions (e.g., text classification) and analyzed for insight into psychological constructs in language.

Word2vec and GloVe: Local Word Co-Occurrence in Large Corpora

The celebrated *word2vec* method (Mikolov, Chen, et al., 2013) was a breakthrough success in using neural networks to learn representations of words, and motivated a great deal of further research attempting to map words to continuous vectors in an "embedding" space. Mikolov, Chen, et al. (2013) trained a feed-forward neural network to predict a word's context from the word itself, referred to as the *skip-gram* model.[33]

[33] Note that two methods were proposed in this work, "Continuous Bag of Words" (CBOW) and the skipgram. We focus on the skipgram model given its further development, and success, in (Mikolov, Sutskever, et al., 2013)

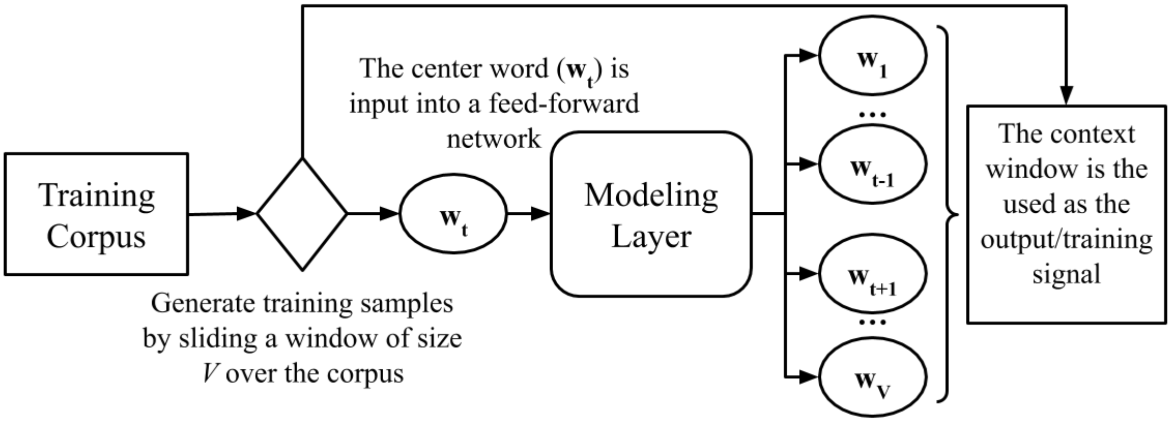
Training Corpus
The center word ($\mathbf{w}_t$) is input into a feed-forward network
Generate training samples by sliding a window of size *V* over the corpus
$\mathbf{w}_t$
Modeling Layer
$\mathbf{w}_1$
$\mathbf{w}_{t-1}$
$\mathbf{w}_{t+1}$
$\mathbf{w}_V$
The context window is the used as the output/training signal

Figure 5. The training procedure for generating word embeddings with the skip-gram algorithm.

The learning objective of the *skip-gram* model is to approximate the probability $p(w_{context}|w_t)$, which is the distribution of a word's context, given the occurrence of the word. The process of fitting this distribution from data is challenging, and requires *approximation* rather than exact derivation. This is the role of the *skip-gram* algorithm (pictured in Figure 5). Given a large text corpus, training samples are generated by a "sliding window" (i.e., considering the words before, and after, the main word, and subsequently moving the window to measure the next main word) of some adjustable size *V* over all the sentences in a corpus (e.g., a corpus of hundreds of thousands of books in English). The middle word of the window is taken as input, and the surrounding words within the sliding window are taken as output. Words (the circular shapes in Figure 5) are represented as dummy-coded variables (typically called "one-hot" variables) covering the entire vocabulary. The modeling layer, which is a 1-layer feedforward network, learns representations of words that are able to approximate $p(w_{context}|w_t)$, yielding word embeddings as the rows of the modeling layer. There are other important aspects of training a *skip-gram* model, however those details except the level of insight we are able to give here.[34]

The general concepts highlighted by the *skip-gram* model are (a) its operationalization of the distributional hypothesis, specifically by using the contexts of words to determine their representation, and (b) the usage of neural networks to model latent, non-linear relationships among words, coupled with large datasets of unlabeled text. In some cases, new *skip-gram* models are trained to datasets in order to analyze a particular corpus in terms of its word relationships. However, most applications typically use other, more developed models, such as "Global Vectors for Word Representation" (GloVe), or variants of *skip-gram* that use the key innovative concepts.

GloVe is a model for word vectors that captures the same contextual information as *skip-gram*, but with greater attention to "global" co-occurrence statistics. Whereas *skip-gram* only ever attends to the words within a given training window, GloVe resembles matrix factorization approaches (e.g., LSA) in attempting to compress all word co-occurrence information into the same vectors (Pennington et al., 2014).

Both the *skip-gram* model and GloVe attempt to learn from words' co-occurrence information. The primary difference, which is argued by the architects of GloVe to provide a more accurate estimate of the same probabilities (between word and context), is the effective usage of global trends. In essence, the skip-gram model was *too* local, in that it optimized one training batch (group of word-context pairs) at a time. Due to the performance improvements by the GloVe model, GloVe vectors have been more commonly used in downstream analysis.

Other Word Embedding Models

[34] A tutorial of the skip-gram architecture is given by Chris McCormick at Word2Vec Tutorial - The Skip-Gram Model · Chris McCormick

FastText. Though several novel methods have been proposed for improving word embeddings, the *FastText* embedding method stands out. Bojanowski et al. (2017) generalized the *skip-gram* loss to character n-grams, allowing for more efficient training and the ability to produce word embeddings for words that were not in the training vocabulary, by computing the embeddings of sub-components of the word in question and averaging them together. This approach had the high-level goal of modeling "sub-word" information in words, which is where we get the concept of a character n-gram. A character n-gram is a sequence of characters with length between 2 and 6, such that a word is a composite of many character n-grams.

The algorithm was used by the authors to train word embeddings on a wide variety of languages, giving researchers the ability to download and analyze word vectors in 157 different languages.[35] These vectors have not been extensively validated, and come from the "common crawl" corpus, which was compiled from online sources. The *fastText* framework can be accessed via the *fastText* python library.[36]

Table 5.

Three prominent word embedding methods

	Description	Contribution & Usecases
skip-gram	From each word, estimate the probability of all other words occurring "near" the initial word.	Pioneered the practice of learning embeddings from local co-occurrence. Used in many previous psychology studies for analyzing word relationships.
GloVe	Reconstruct (factorize) a matrix containing each word-word co-occurrences within a sliding window.	Captures more "global" context, and thus has largely replaced *skip-gram*.
fastText	Skigram generalized to character	Faster than *skip-gram*, thus

[35] https://fasttext.cc/docs/en/crawl-vectors.html
[36] https://github.com/facebookresearch/fastText

	n-grams. Computes word embeddings by aggregating their character *n*-grams ("quick" = "qu" + "ui" + …)	easy to fit to new data. Can calculate embeddings for unfamiliar words. Available for download in many languages.

Word Embeddings in Analysis

There are generally three ways in which word embeddings can provide information on psychological phenomena: (1) to quantify cultural, psychological, and other constructs across corpora, languages, and time by using geometric relationships (e.g., the distance between two points in a coordinate system) among words in the embedding space, for example the distance among vectors; (2) to use them as features in a supervised prediction task, such as predicting humans' judgments of words (or word categories) based on those words' embeddings; and (3) to train from scratch in order to analyze measures from embeddings, such as those in (1) and (2), across multiple corpora.

Extracting information from the geometry of an embedding. Analyzing the semantic space provided by word embeddings is a practice initiated with earlier distributed semantics methods, in particular LSA. The relatedness among words in a continuous space, measurable through geometric manipulations, can tell stories about the corpus, the authors of the text, the time period, and even the particular language or language community. Using geometric relations in semantic space to infer relationships has precedent in psychology, where LSA can be used in a similar fashion.

Distributed dictionary representations (DDR; Garten et al., 2018) apply this reasoning by averaging the word vectors from dictionaries, thus representing a dictionary in word embedding space. A dictionary's "center" could then be compared to the embeddings from natural language text, such as social media posts. Instead of counting dictionary words in a text, the DDR method determines the semantic similarity between the words in the text and the words in the dictionary, which increases the accuracy of the analysis especially for short texts.

Other than the geometric similarity between word vectors, another analytic technique is to use vector addition and subtraction to examine the space between constructs in an embedding space. In their original paper on the *skip-gram* architecture, (Mikolov, Sutskever, et al., 2013) found a compelling relationship in the word vectors: additions and subtractions produced meaningful transformations of different types and forms of words. For example, the vector operation "king" + "woman" - "man" (which produces a vector) is highly similar, or close, to the vector for "queen." Understanding this to mean that these vector operations are able to capture abstract "dimensions" of language, such as the "gender dimension," Kozlowski et al. (2019) used word embedding-based measures to study cultural dimensions, such as gender, class, and race.

Predictive tasks. Using similarity, vector addition, or other geometric measures from embeddings directly draws information from the embedding space, yielding correlational measures that can show differences among semantic categories (e.g., gender and occupation; Lewis & Lupyan, 2020). Similarly, word embeddings can be used in predictive models, providing evidence of the form "word embeddings can predict phenomenon x."

For example, Richie et al. (2019) tested whether *skip-gram* word vectors (as well as a host of alternative embedding methods such as *fastText*) are able to predict 14 human judgments of words and phrases from a diverse selection of behavioral domains. Judgements included whether a trait was masculine or feminine, whether a brand was sincere and exciting, and the significance of certain occupations. Embedding vectors of each of the target words was computed using pretrained models, and these vectors were used in a regularized regression of the average human judgment of the corresponding target. The authors found that models' predictions were highly correlated with humans' judgments, showing that word embeddings contain enough information to predict these high-level dimensions of entities. Importantly, the interpretation of these results is slightly different than just examining similarities among words, as a linear transform of the embeddings (i.e., the regression) was necessary to establish the degree of relatedness between embeddings and human judgment.

For predictive tasks involving naturally occurring text (as opposed to word lists, as in the example above with human judgments) the embeddings of words in a sentence/document can be averaged together, such that the sentence/document is represented as the midpoint of the words in the embedding space. This is highly similar to the DDR method described above, which compared midpoints of documents' embeddings to the midpoints of dictionaries. In this case, the average/midpoint of a document's word embedding can be a highly informative baseline for prediction tasks, in most cases outperforming TF-IDF (Joulin et al., 2017). However, as we will see, more sophisticated approaches can be used to generate representations of documents.

Comparisons across corpora. In certain cases, pretrained word embeddings are insufficient for a modeling task. When two or more corpora are to be compared, word embedding models can be trained for each corpus and inferences subsequently made based on differences in the relationships among words. For example, Garg et al. (2018) trained separate GloVe models on corpora corresponding to each year of New York Times articles. By tracking the similarities among word sets over time the authors found that attitudes towards women and ethnic minorities corresponded to independent indicators of public opinion over time. In general, training new word embeddings is more costly than reusing them; however, doing so can be appropriate when the requisite data are available (i.e., enough text) and the research questions are not addressable using existing embeddings.

Closely related to this analysis technique is to incorporate covariates of interest directly into the word embedding model, via “dynamic” word embeddings. Researchers developed and used this technique to study “semantic drift,” the temporal change in the meaning and usage of words (Bamler & Mandt, 2017). Word embeddings were fit similarly to the *skip-gram* architecture, but with time-stamp information included as a covariate.

Text Encoding: Mapping Word Sequences to Continuous Space

We have seen how embedding methods can capture the meaning of words, and how word-level meaning can be used in analysis. Sentence-level (or in general, sequences of words, including paragraphs, social media posts, or entire documents) representations are similarly useful, though they are, understatedly, even more difficult to produce. With these representations, we can analyze meaning in continuous space (similarly to the approach to analyzing the geometry of word vectors) and generate even better predictions. We are already familiar with baseline approaches to this task: TF-IDF vectors, for example, express sentences and documents numerically such that statistical analyses may follow. The opportunity of more sophisticated approaches in comparison to word counting is, essentially, to greatly improve the “correctness” of our model of language data, allowing more sophisticated analyses and better predictions. But, to reiterate our discussion of the “curse of dimensionality” in the introduction to this section, actually modeling text in this way is extremely challenging, even for modern methods. There are simply too many ways that words can be combined to generate meaning (Chomsky, 1965/2014), and not enough data such that we can observe every combination. What we need, and what neural networks have begun to show a propensity for, is capturing general features of language, specifically how words are combined in order to make the meanings we operate with every day, from data.

Counting methods, such as dictionaries and LSA, probabilistic bag of words models like LDA, and even word embedding models, ignore the sequence of words. This assumption that the ordering of words is inconsequential (or rather, that attempting to model word order is too difficult, given the explosion in possible combinations of words into word sequences) can often be limiting. Clearly, this practically minded approach ignores the role of syntax and the fact that humans in fact communicate with sentences, and not bags of words. Neural networks have been shown to be able to capture the dependencies in language that are due to word order and "compositional" semantics (Baroni, 2020). By composition, we mean the fact that words are combined to produce meaning (as we know it, at the sentence level and beyond) via the rules of the particular language, rather than just being grouped together or observed in association (Partee, 1995). Though neural networks might not *truly* learn composition as humans know it, they have been demonstrated to contain *some* features of a compositional understanding of language (Hupkes et al., 2020).

We will highlight three types of text encoding methods that are useful in the applied setting, each of which has variations that can be used for both prediction (i.e., text classification or regression) and producing general-purpose embeddings of sentences. First, a variety of techniques have extended the logic of the *skip-gram* architecture, specifically the relationship between a word and its context, to the sentence and paragraph level. Methods like *doc2vec* (Le & Mikolov, 2014) were initially proposed directly along these lines, and later the *skip-thought* model (Kiros et al., 2015) embedded sentences based on the relationship between a given sentence and its surrounding sentences. Second, neural networks that are constructed to handle sequential data (e.g., Recurrent Neural Networks) can be used to obtain general representations of text (Conneau, Kiela, et al., 2017; Peters et al., 2018) as well as effectively be used to learn a text classifier or regression. And third, the "transformer" architecture, and more specifically the practice of "fine-tuning" previously trained models to new datasets, has become the leading technique in NLP for encoding text for most modeling tasks.

Table 6.

Summary of Neural Network Models for Text Encoding

	Description	Applications
Average Word Embeddings (e.g., Pennington et al., 2014)	Map words in text to embeddings, average them to produce a single embedding	Baseline features for text classification and prediction that make use of word embeddings
Doc2vec (Le & Mikolov, 2014)	Previously standard method to use document context (i.e., paragraph sequences) to improve text embedding	Can be trained from scratch to get document embeddings from a corpus.
skip-thought vectors (Kiros et al., 2015)	Uses an RNN model to extend the *skip-gram* architecture to sentences	General sentence encoder. Can generate vectors of sentences
Recurrent Neural Networks[37] (Elman, 1990; Mikolov et al., 2010)	Used to process words in sequence. Variants include LSTM, bi-LSTM, GRU	Supervised modeling, such as text classification, uses RNNs to model sequential dependencies among words.
InferSent (Conneau et al., 2017)	A bi-LSTM network predicts the entailment of two sentences	General sentence encoder.
ELMO (Peters et al., 2018)	Contextual Word Embeddings from a bi-LSTM language model	Generate contextualized word embeddings (of words in naturalistic text)
BERT (Devlin et al., 2019)	*Transformer* architecture trained to predict masked words and sentence ordering	Generate contextualized word embeddings; fine-tune to supervised tasks

Extending *skip-gram*: *Doc2Vec* and *skip-thought* vectors

[37] Similarly, a CNN can be used for supervised modeling over sequences of words (Kim, 2014)

Though methods in NLP quickly fall out of fashion once they have been surpassed in performance by a newer, more sophisticated method, the past standards can be instructive, especially since current methods continue to reuse previous ideas in more efficient, successful ways. One way that researchers modeled text at the document level is to consider the surrounding contexts of sentences and paragraphs. The *doc2vec* method was an early attempt to accomplish this, and extended Mikolov et al. (2013) to additionally model paragraphs within a sliding window (Le & Mikolov, 2014). As an example application of doc2vec, Dehghani et al., (2017) trained models on corpora in different languages (English, Mandarin, and Farsi) in order to test whether fMRI data, collected from participants reading stories in those languages, could be predicted by the doc2vec representation of the given story.

Similarly, *skip-thought* vectors (Kiros et al., 2015) are derived from the *skip-gram* training paradigm. The *skip-thought* architecture takes a given sentence as input, encodes this sentence into a vector and then tries to generate (in sequence) the previous and subsequent sentences. In this way, a representation of a sentence will be forced to model the aspects of text that tell us about the sequencing of sentences. Once trained, the skip-thoughts model could be used as a “sentence encoder.” Essentially, any sentence could be taken as input, and the pre-trained *skip-gram* model would output a fixed-length vector that contained the models’ knowledge of sentence ordering. This modeling approach (encoding sentences into vector representations) has become a standard practice in NLP, with the methods merely becoming better.

Modeling Words in Sequence with RNNs

Recurrent Neural Networks (RNNs; Elman, 1990) are the neural architecture that is most naturally suited to modeling language. RNNs are suited to modeling the dependencies among words, such as the type of dependencies that n-grams attempt to quantify (that "year" often follows the words "happy new"), as well as longer range dependencies that are characteristic of grammar in most languages. In practice, plain RNNs are difficult to fit to data, so two variants are commonly used: Long Short-Term Memory (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (Cho et al., 2014). The most common usage for RNNs is in supervised tasks, for example document classification, where their ability to detect patterns in sequences of words is effectively used to produce high-performing predictions. Additionally, the RNN architecture is the basis of other text encoding methods, and will likely continue to influence the future of NLP.

Text classification with RNNs. One common way that RNNs can be used in text analysis is in text classification.[38] These methods have three components: a word embedding layer, sequence modeling, and an output layer. First, each word is mapped to a vector representation from an embedding matrix, with one row per unique word in the vocabulary and one column for each feature in the embedding space. Pretrained embeddings, such as GloVe or *fastText*, can be used to construct this embedding matrix; the alternative is to initialize this matrix to be random and train word representations from scratch. In most cases, using pretrained word embeddings in conjunction with an LSTM provides an advantage over simply averaging the word embeddings, as the sequential relationships among the word embeddings is taken into account. An example of this approach is Mooijman et al. (2018), which used LSTM networks to classify moral rhetoric in Twitter posts, with the first embedding layer initialized using pretrained GloVe vectors.

Additional factors have to be taken into consideration when training networks, compared to models like SVMs. This has to do with the fact that neural networks are more volatile than other algorithms, and that optimizing their predictive abilities often comes down to a set of

[38] Text regression and classification in this setting are interchangeable.

engineering practices and heuristics. The interested reader is advised to investigate this separately (e.g., instructive books include Géron, 2019). Here, we review several key concepts and techniques. First, “dropout” layers can be added to any non-output layer in a neural network, and zero-out, or “drop out,” a random percentage of the layer’s weights on each training iteration. Second, one should set aside a validation set (separate from a “test” set) to use for early stopping, which is stopping training iterations only when the loss on the validation set has essentially leveled off. While cross-validation is possible for these models, it is much more costly, as neural networks take quite a bit longer to train as traditional statistical approaches. Thus, a test set (typically 20%) is designated, and a separate validation set is used to tune hyperparameters.

Multi-task and multi-input networks. One promising analytic technique is to multi-task and multi-input networks to evaluate complex relationships among language-based variables. Multi-task learning is when one is training a network on multiple objectives, representing distinct information in each task, such that the representation learned by the network must predict both tasks simultaneously. Multi-task networks can provide predictive benefits (multi-task networks can sometimes learn more efficiently than single-task, which is the default). For example, Hoover et al. (2019) used multi-task LSTM networks to jointly predict 5 moral sentiment labels per social media post. In this case, the LSTM was responsible for learning all 5 classifications jointly, while each classification had a distinct output layer.

Multi-input networks enable multiple modalities of input signal for a single prediction task. In language analysis, this can mean adding additional predictors (such as metadata corresponding to a document) to a neural network. This can allow inferences to be made about the effect of those metadata on the performance of the text prediction task, such as classification.

For example, in considering the three-way relationship between language, background knowledge, and political rhetoric, Garten & Kennedy et al. (2019) compared models with and without access to background knowledge in order to infer whether political rhetoric was variably detectable from text. They found that, by comparing LSTM networks, some of which had one-hot encodings of background information (e.g., the politician's political party) concatenated to the feature representation of the text, a significant improvement in prediction performance was observed when the LSTM network had access to background information.

Generating text embeddings with sequential models. Just as with prior methods, such as doc2vec, *skip-thoughts*, or even TF-IDF, RNN-based models can be used to produce continuous representations, or embeddings, of text. In other words, RNNs do not apply just to classification. Here, we will highlight two methods that have already become overshadowed by more recent methods; however, they are again instructive, and have already been used for text analysis.

First, the InferSent model is a text encoder that provides accessible embeddings of sentences in English[39] (Conneau, Kiela, et al., 2017). Rather than just learning patterns from large corpora of unlabeled text, InferSent functions by training a multi-input LSTM network (a variant of an RNN architecture) to predict the "entailment" relation between two sentences: implication (one implies the other), contradiction (one implies the opposite of the other), or neutral (neither implication nor contradiction). In the process of learning to model this aspect of language, the model was able to distill important aspects of syntax and composition, yielding a trained model that could generate informative embeddings of sentences with no additional training.

[39] https://github.com/facebookresearch/InferSent

InferSent can be used to generate embeddings for new text without any new training, similarly to how word embeddings can be plugged in to a new dataset without new training or modeling. Such encoding models can be used in various text analysis settings to great effect. For example, Atari et al. (2020) used InferSent to produce embeddings of moral foundations vignettes (Clifford et al., 2015), which were then used to predict how individuals' reported their bodily activations in response to those same vignettes.

Another technique for generating text embeddings, which similarly uses an LSTM-based neural architecture, is Embeddings from Language Models (ELMO; Peters et al., 2018). ELMO established several key concepts that have become central in NLP methods. One of the motivations of ELMO was to provide word embeddings that were based on the context of the particular sentence, rather than map a word to the same embedding every time. Their success yielded what they call "contextualized word embeddings": given a sentence, each word is mapped to an embedding that is dependent on the particular context of that sentence. Thus, polysemous words (those with multiple meanings) mapped to different vectors depending on the observed context. For the first time, ELMO was able to produce a semantic space that was based on words in usage, rather than words as singular, irreducible entities.

ELMO has seen several applications in psychology, primarily as a means to construct contextualized embeddings. For example, Richie et al. (2019) included ELMO word embeddings for their task of predicting human judgments of words, by combining judgment target words with words for the categories to which they belonged (e.g., "bass food"). But in reality, ELMO did not see widespread adoption in text analysis despite its groundbreaking status, as it was quickly succeeded by a new wave of models that improved contextualized word embeddings (and more groundbreaking areas), using the same basic idea. This is the subject of the following section.

Pretrained Transformer Language Models

NLP has changed radically in the past several years with the rising prominence of "transfer learning," which is the practice of applying, or reusing, knowledge from one task or domain to another task or domain (Torrey & Shavlik, 2010). In NLP, the practice of transfer learning gained significant momentum with the development of word embeddings, which could be applied in new analyses and supervised modeling without "relearning" the representation of words, and text embedding methods such as InferSent, which allow highly informative text embeddings to be produced by leveraging what had been learned by a model in a previous task. Recently, NLP has seen a wave of new methodologies centered around learning and transferring exceedingly sophisticated models of language. Above, we saw how ELMO could be used to generate contextualized word embeddings from a bidirectional LSTM that used learning over sequences. Similarly, Howard & Ruder (2018) achieved impressive text classification performance by fine-tuning an entire three-layer LSTM model to smaller-scale tasks.

These initial innovations in transfer learning were later improved upon, both in terms of producing word and text embeddings and in supervised tasks, by using a new neural architecture, the Transformer (Vaswani et al., 2017). With this architecture, The Bidirectional Encoder Representations from Transformers model (BERT; Devlin et al., 2019) was a breakthrough achievement in transfer learning for NLP, and much of the relevant tools for text analysis are based on BERT and its variants. In this section, we will exclusively focus on BERT in terms of explaining how it works and how it is used, and give resources for using its variants.

The architecture used by BERT consists of many "attention" layers (12 in the case of the "base" model, 24 in the case of the "large" model; both were trained and provided for download by Devlin et al., 2019). Attention in this setting means that the layer must attend to relevant words in the input.[40] Relevance, in turn, is defined by the model's objective function: masked-word prediction, which involves predicting the identity of randomly masked tokens and whether two sentences are sequentially ordered (i.e., one follows the other) or not. With these two learning objectives, an incredibly powerful modeling architecture, and an inordinate amount of training over billions of words of input, BERT models learn to encode linguistic structure far beyond its predecessors.

For the purposes of understanding this type of model intuitively, and its successes in terms of its modeling objectives, we can compare BERT to earlier models. The *skip-gram* model, for example, predicting a word's context given the occurrence of the word, and the skip-thought model learned by predicting the sequence of sentences. We can see both of these concepts in the BERT objective: we predict some words given other words in the sentence, and we model the sequence of sentences. The difference is in the increased complexity of the word masking prediction, the increased model complexity of the multi-layer Transformer architecture, and the intense effort of training. Though BERT and similar techniques are understandably celebrated as the next major wave of NLP methodologies, it is beneficial from the practitioner's perspective to consider that BERT succeeded by more successfully optimizing similar ideas in the recent history of NLP research.

Generating embeddings with pretrained language models. Just as the ELMO model was used to generate contextualized embeddings and embeddings of text based on pretrained language models, Transformer models like BERT can be used to generate embeddings. In most cases, these embeddings result in predictive performance benefits over ELMO. Unfortunately, as of the present writing the process for generating embeddings from a model is much more involved than for word embeddings. Whereas GloVe vectors (for example) can be downloaded

[40] Alammar (2018) provides a description of the Transformer architecture, specifically how "attention" is used

and used without any additional hassle, embeddings require models to be loaded, text to be preprocessed and tokenized, and model outputs to be recorded and aggregated. This is because BERT and similar models yield contextualized representations: GloVe vectors are a mapping from unique words to single vectors, whereas BERT word embeddings are computed dynamically for each occurrence of a given word.

Though in the future a more streamlined solution might be available for working with BERT, currently we are limited to a multi-step process in the Python programming language. The best process currently for working with BERT is to use the *tokenizers*[41] library in Python to tokenize inputs and the *transformers*[42] library to download pretrained models and extract embeddings from tokenized inputs. With this pipeline — outlined by McCormick (2019) — one can extract contextualized embeddings (specifically, a list of word vectors for each word in the input), or sentence embeddings (word vectors are averaged, much as GloVe vectors could be averaged for a sentence).

Fine-tuning for supervised modeling. Fine-tuning is the practice of training an existing model on a new dataset, such that the existing model is largely the same but “tuned” to the particular supervised task. Fine-tuning a model such as BERT involves taking a pretrained model and tokenizing inputs just as outlined above. The additional step with fine-tuning is that labeled data, most often a classification label for applications in the social sciences, are used as supervision for a full BERT model with its weights initialized to that of the previously trained model. Conceptually, fine-tuning a BERT model is similar to training an LSTM that has had its first layer initialized with word embeddings; the difference is only that the entire multi-layer model for BERT has been initialized. Here, all the same training principles apply which were

[41] https://github.com/huggingface/tokenizers
[42] https://huggingface.co/transformers/

described for training and validating models, including using held-out data to tune parameters, a separate "test" set to report final model performance.

Discussion

It is hard to overemphasize the dynamic nature of the leading edge of text analysis. Because NLP (and by extension, machine learning) continues to change so rapidly, staying ahead of the curve is not a viable option. Instead, here we have reviewed and instructed on the application of these novel methods in analytic contexts. Conceptually, knowledge of word embeddings (e.g., *skip-gram* and GloVe) and tunable language models (e.g., BERT) is a reusable store of practical knowledge that the inclined reader will undoubtedly employ in the future. The particular concepts that we believe to be of greatest importance are: (a) that models are increasingly data-driven, which pushes language-related scholarship towards whatever data are available; (b) that models are defined in large part by learning to predict relevant phenomena, predominantly linguistic (e.g., word order or annotated meaning); and (c) that, for leading methods to be used effectively in psychology, psychological aims and theories ought to be integrated more cohesively with NLP and machine learning research. Below, in our general discussion, we touch on several methodological trends in NLP and deep learning that are worthy of the reader's attention.

General Discussion

Language analysis has changed dramatically over the past decade due to the influx of new technologies, interested parties, and rich datasets. It will likely change similarly in the next ten years, due to the same factors. However, through the course of language analysis in psychology there has been a unifying thread: the integration of theory and method. In this chapter, we aimed to provide a template for language analysis that will continue to be informative and useful amidst the coming changes in NLP, document and review current practices and techniques, and build intuitions into the more complex areas of the new wave of methods. We conclude with prognostication, anticipation, and several words of warning.

What Will Stay the Same?

First, a variety of the techniques and practices will likely remain in text analysis for the foreseeable future. The lexicon approach of dictionaries — applying theoretical knowledge to words via word lists — is probably here to stay, though we can expect an evolution in how dictionaries are built, applied, and interpreted due to the increasing information yielded by computational techniques, such as word embeddings. Specifically, word embeddings help us to realize more fine-grained categorizations of words, and may prove invaluable for tuning and modifying word categories. At the same time, the new wave of contextualized word embeddings promises the dynamic application of word embeddings, such that words' multiple contextual meanings can be appreciated in dictionary-based measures. Second, though manual data annotation is far from a refined process, and labor-intensive at best, it is an important practice for researchers to engage in, both in terms of validating automated measures as well as in clarifying how exactly theoretical constructs map to language. Much of the work of developing typologies is in proposing, refining, and sometimes rejecting coding rules; this process inevitably improves the conceptual clarity of target constructs in language, particularly for subjective phenomena of interest in psychology.

Third, practices and concepts drawn from traditional machine learning — model validation, regularization, supervised versus unsupervised learning — are probably not going to go out of style. Indeed, these facets of statistical learning are well-established, and ought to become increasingly used in psychology. Finally, though we cannot say definitively that deep learning and neural networks are here to say, the concept of transfer learning probably is. In various ways, transfer learning has taken over the applied data analysis world. It is a simple fact that so many human phenomena, especially language, are difficult to model from scratch. In NLP, pretrained language models are currently the rage, and the next generation will probably be different in subtle though impactful ways; however, it is likely that the new waves of models will still rely on learning *and transferring* knowledge. Below, we also touch on the steps that psychologists can make to maximize the potential of this paradigm towards analysis, rather than prediction.

Looking to the Future

NLP, as a field, has become driven by successful prediction and performance in the wild. For example, it is obvious that a machine translation system that can translate better is to be desired. This goes in hand with the understanding of “good” representations outlined by Bengio et al. (2013): a good internal representation inevitably leads to good predictions. However, it is not yet clear how this prediction-oriented discipline manifests in a clear path to better theory-driven language analysis.

The core problem holding back this multidisciplinary cooperation is that neural networks excel at prediction yet do not directly increase our knowledge of the modeled phenomena. As a case in point, consider studying the relationship between individuals’ language and their personality attributes. In this setting, neural network models will provide more accurate measures of the *magnitude* of the relationship — how related are personality and language — but do not inherently shed light into the nature of the relationship. Indeed, bag of words models have been effectively used for just this purpose, though potentially less accurately (given the inferior quality of representation). Currently, the most exciting and uncertain direction of NLP research is in devising techniques for analysis that build on the successes of neural networks. This area is, more or less, focused on the interpretability of neural networks.

In some cases, interpretability is discussed within the framework of trust, specifically between machines and those who operate them. For example, medical professionals using a machine learning algorithm to parse individuals' medical histories might distrust a set of predictions if they are opaque and uninterpretable, because the chain of reason by which a prediction is reached is more important than the prediction itself. A sizeable literature in machine learning and natural language processing has emerged in recent years that attempts to develop explainable deep learning algorithms for language (called 'intrinsic' explainability; Murdoch et al., 2018) or to apply post hoc methods to produce explanations of model behavior (Jin et al., 2019; Singh et al., 2018). In contrast to many of the assumed settings in NLP and machine learning research, interpretability in language analysis means that a network's predictions can be understood given an input, such that a predefined construct can be explored or a hypothesis tested. For the most part, this type of interpretability research has yet to be fully developed.

One promising type of interpretable machine learning is post hoc explanation. In this setting, post hoc indicates that how models represent data internally is not taken into consideration; rather, only models' predictions are considered. For example, classifiers can be evaluated and corrected for undesirable behaviors using post hoc techniques, even with uninterpretable models such as BERT (Kennedy & Jin et al., 2020).

What are the Problems?

NLP is English-centric

Models need massive text datasets, and massive text datasets are more available (or more frequently compiled) in English. This is a simple fact that inhibits NLP research, and accordingly any research that applies NLP techniques. Some research attempts to build resources in multiple languages (Bojanowski et al., 2017) but many works in NLP assume English as the default language, and English speakers as the default language community. It has gotten to the point where merely declaring which language a study or method is performed in has become the focal point of reformist efforts (Bender, 2019). Recognizing this limitation of NLP research is critical for psychologists using NLP techniques, as failing to do so may overlook confounds that reduce the generalizability of studies.

Beyond raising awareness, this issue is also a call to psychologists to contribute to the diversification of language analysis tools and studied populations. This language imbalance is similar to that observed in psychology studies, which suffer from a bias towards Western, Educated, Industrial, Rich, and Democratic societies (WEIRD; Henrich et al., 2010) and thus require concerted efforts to be more inclusive and representative. Psychologists working in language analysis can design tools (e.g., new dictionaries in non-English languages, annotated datasets in non-English languages) and help to validate those tools with non-English-speaking participants. One key to the success of such ventures is the inclusion of non-WEIRD researchers (i.e., psychologists fluent in the language of a given study), rather than relying on purely automated methods for translation, such as Google translate, which introduce myriad confounds that have yet to be measured in dedicated studies.

Models in NLP are Biased

Pretrained models, including word vectors and full models such as BERT, can be biased against protected (i.e., marginalized) groups (e.g., LGBTQ identifying persons, minority ethnic groups, etc.). These biases typically manifest as disproportionate error rates of classifiers on inputs from certain disadvantaged communities, and models and representations reinforcing negative stereotypes. Some high-profile examples of this include (Bolukbasi et al., 2016; Caliskan et al., 2017), who showed that popular word embeddings encode gender and racial stereotypes (see also Chapters 24, 25, and 26 in the present volume).

Why are models biased? The answer is incredibly complex and cannot be boiled down to a few root causes; indeed, if this were so, the field of AI and machine learning would have little controversy in establishing a plan for dealing with bias, which is not so.[43] Some pieces of the puzzle, which are relevant to the techniques introduced in this chapter, are problematic (and unexamined) training datasets, as well as large, uninterpretable neural models which are likely to capture biases from data in unintended ways (Bender & Gebru et al., 2021). One of the faults of networks is that they are highly data requisite. This is not inherently restrictive, as transfer learning approaches (e.g., word embeddings, fine-tuning) can be used for smaller datasets. The problem lies in the fact that the text data that models are trained on is misunderstood and problematic. Before the Internet, social media, and digitized libraries, it is possible that deep learning would never have worked. Models in NLP today rely on massive text datasets that have been collected automatically from such sources as Wikipedia, the Common Crawl,[44] books corpora, and online comments. However, these large text datasets carry with them certain issues, one of which is that they are largely unexamined in terms of sources (e.g., social media users and online forum commenters) and content (e.g., containing social biases against women and non-white minorities, which are present throughout published texts; Paullada et al., 2020). As a result, powerful neural network models capture gender bias, racial bias, and outright prejudice in some cases. In addition, as discussed above, the lack of interpretability of neural networks

[43] See, for example, this synopsis on the controversy emerging following the publication of a computer vision system that reconstructed pixelated images of faces in a racially biased manner (Kurenkov, 2020)

[44] https://commoncrawl.org/

compounds the issue, as researchers and users of pretrained models cannot grasp the depth and diversity of model biases.

Bias reduction has become a sizable literature in machine learning (Mehrabi et al., 2019). For example, gender bias has been demonstrated in word embeddings, as the association of female words and pronouns with “homemaking” and other wrongfully stereotypical occupations, and male words and pronouns with roles including business, computer programming, and science. To combat these biases, researchers have proposed applying transformations to the word embedding space that isolate the “gender” dimension, and attempt to debias it (Bolukbasi et al., 2016). However, recent research has also shown that these debiasing techniques are mostly ineffective, and merely “cover up” biases rather than remove them (Gonen & Goldberg, 2019). This is a reasonable indicator of bias research today: it is a problem, we have some solutions, but largely it is an open question as to how we can totally remove bias from models that are trained on inherently biased data. To the reader interested in reducing the negative effects of bias in language analysis, our recommendation is to proceed with caution, recognize the effects of bias, and continue to build interdisciplinary collaboration with NLP researchers combating bias. Though it requires a greater investment of time and resources (and indeed can be impossible with today’s neural network methods), embeddings can be retrained from “unbiased,” or unproblematic, datasets.

To the reader interested in studying bias in NLP rather than taming it (though these practices are obviously linked), one merely has to understand the basics of how information is extracted from embeddings (i.e., continuous representations) in order to measure bias constructs. By doing so, psychologists can contribute to the literature in NLP on bias and fairness, helping to inform other researchers of new problems with certain models and new ways in which human biases influence computational models.

An Alarmingly Accelerating Carbon Footprint

Because of the direction of NLP and, more broadly, the neural network-centered research agenda of AI and machine learning, the deep models that drive innovation and insight are becoming so expensive to train that they are producing a sizable carbon footprint. In NLP, recent models such as BERT and Transformer architectures have been shown to have a significantly larger carbon footprint than that produced by a car over its entire lifetime (Strubell, Ganesh, & McCallum, 2019). In short, though current technologies may enable us to train massive models, thus achieving remarkable performance, the cost is still very much there. From the perspective of language analysis, there is extra incentive not to develop research frameworks that depend on prohibitively expensive computational costs. Luckily, the trend in NLP in recent years, which widely promotes the distribution of pretrained models such that models rarely need to be retrained (Wolf et al., 2020), ought to benefit those interested in using NLP technologies for analytic purposes. However, the efficiency and computational cost of a model ought to be viewed as a basic criteria for using that model for downstream purposes (R. Schwartz et al., 2020). In other words, those interested in using deep learning models for language analysis can review models' performance on certain tasks as well as their efficiency and carbon footprint, placing a higher emphasis on the latter. This should not, however, prevent researchers from using already-trained models, but can merely provide a better awareness of the cost and benefits of deep learning.

Parting Thoughts

This chapter contains a comprehensive view on the methods available for language analysis in psychology, both in terms of high-level insights and implementation. Understatedly, the world of text analysis extends beyond what we could deliver here, particularly in the direction of NLP research and content coding in the social sciences. However, we believe this chapter will stand the test of time, for though the particular methods might evolve and improve, the basic objectives, practices, and concepts will be maintained.

The potential for language analysis in psychology is vast and varied. We hope that, for many researchers, this chapter is merely the beginning, as new questions can be asked, novel datasets constructed and distributed, and more integrative methods developed.

References

Alammar, J. (2018, June 27). The Illustrated Transformer. Visualizing Machine Learning One Concept at a Time. http://jalammar.github.io/illustrated-transformer/

Alizadeh, M., Weber, I., Cioffi-Revilla, C., Fortunato, S., & Macy, M. (2017). Psychological and personality profiles of political extremists. *arXiv preprint arXiv:1704.00119.*

Allport, G. W. (1942). The use of personal documents in psychological science. *Social Science Research Council Bulletin*.

Atari, M., Mostafazadeh Davani, A., & Dehghani, M. (2020). Body Maps of Moral Concerns. *Psychological Science*, 31(2), 160–169.

Baayen, R. H., Piepenbrock, R., & Gulikers, L. (1995). The CELEX lexical database (release 2). Distributed by the Linguistic Data Consortium, University of Pennsylvania.

Back, M. D., Küfner, A. C., & Egloff, B. (2010). The emotional timeline of September 11, 2001. *Psychological Science*, 21(10), 1417-1419.

Bamler, R., & Mandt, S. (2017, July). Dynamic word embeddings. In *International conference on Machine Learning* (pp. 380-389). PMLR.

Baroni, M. (2020). Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 375(1791), 20190307.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238–247.

Bender, E. (2019). The #BenderRule: On Naming the Languages We Study and Why It Matters. The Gradient. https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research: JMLR*, 3(Feb), 1137–1155.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research: JMLR*, 3(Jan), 993–1022.

Bianchi, F., Terragni, S., & Hovy, D. (2020). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *arXiv preprint arXiv:2004.03974*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

Bolukbasi, T., Chang, K. W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 4356-4364).

Boot, P., Zijlstra, H., & Geenen, R. (2017). The Dutch translation of the Linguistic Inquiry and Word Count (LIWC) 2007 dictionary. *Dutch Journal of Applied Linguistics*, 6(1), 65–76.

Boyd, R. L., & Schwartz, H. A. (2021). Natural language analysis and the psychology of verbal behavior: The past, present, and future states of the field. *Journal of Language and Social Psychology*, 40(1), 1–21.

Byrt, T., Bishop, J., & Carlin, J. B. (1993). Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5), 423–429.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183–186.

Carey, A. L., Brucks, M. S., Küfner, A. C. P., Holtzman, N. S., Große Deters, F., Back, M. D., Donnellan, M. B., Pennebaker, J. W., & Mehl, M. R. (2015). Narcissism and the use of personal pronouns revisited. *Journal of Personality and Social Psychology*, 109(3), e1–e15.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., & Blei, D. M. (2009, December). Reading tea leaves: how humans interpret topic models. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems* (pp. 288-296).

Chan, T. W., & Goldthorpe, J. H. (2007). Class and Status: The Conceptual Distinction and its Empirical Relevance. *American Sociological Review*, 72(4), 512–532.

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). BTM: Topic Modeling over Short Texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 2928–2941.

Chen, W., Su, Y., Shen, Y., Chen, Z., Yan, X., & Wang, W. Y. (2019). How Large a Vocabulary Does Text Classification Need? A Variational Approach to Vocabulary Selection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 3487–3497.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder--Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Chomsky, N. (1965/2014). Aspects of the Theory of Syntax. MIT Press.

Cialdini, R. B., Borden, R. J., Thorne, A., Walker, M. R., Freeman, S., & Sloan, L. R. (1976). Basking in reflected glory: Three (football) field studies. *Journal of Personality and Social Psychology*, 34(3), 366.

Clifford, S., Iyengar, V., Cabeza, R., & Sinnott-Armstrong, W. (2015). Moral foundations vignettes: a standardized stimulus database of scenarios based on moral foundations theory. *Behavior Research Methods*, 47(4), 1178–1198.

Cohn, M. A., Mehl, M. R., & Pennebaker, J. W. (2004). Linguistic markers of psychological change surrounding September 11, 2001. *Psychological Science*, 15(10), 687–693.

Collins, S. E., Chawla, N., Hsu, S. H., Grow, J., Otto, J. M., & Marlatt, G. A. (2009). Language-based measures of mindfulness: initial validity and clinical utility. *Psychology of Addictive Behaviors: Journal of the Society of Psychologists in Addictive Behaviors*, 23(4), 743–749.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 670-680).

Conneau, A., Schwenk, H., Cun, Y. L., & Barrault, L. (2017). Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (pp. 1107-1116). Association for Computational Linguistics.

Conway, L. G., Conway, K. R., & Houck, S. C. (2020). Validating automated integrative complexity: Natural language processing and the Donald Trump Test. In *Journal of Social and Political Psychology* (Vol. 8, Issue 2, pp. 504–524).

Coppersmith, G., Dredze, M., & Harman, C. (2014, June). Quantifying mental health signals in Twitter. In *Proceedings of the 1st Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality* (pp. 51-60).

Coppersmith, G., Dredze, M., Harman, C., & Hollingshead, K. (2015). From ADHD to SAD: Analyzing the language of mental health on Twitter through self-reported diagnoses. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, 1–10.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273–297.

Dean, H. J., & Boyd, R. L. (2020). Deep into that darkness peering: a computational analysis of the role of depression in Edgar Allan Poe's life and death. *Journal of Affective Disorders*, 266, 482-491.

DeAndrea, D. C., Shaw, A. S., & Levine, T. R. (2010). Online Language: The Role of Culture in Self-Expression and Self-Construal on Facebook. *Journal of Language and Social Psychology*, 29(4), 425–442.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391–407.

Dehghani, M., Bang, M., Medin, D., Marin, A., Leddon, E., & Waxman, S. (2013). Epistemologies in the Text of Children's Books: Native- and non-Native-authored books. *International Journal of Science Education*, 35(13), 2133–2151.

Dehghani, M., Boghrati, R., Man, K., Hoover, J., Gimbel, S. I., Vaswani, A., Zevin, J. D., Immordino-Yang, M. H., Gordon, A. S., Damasio, A., & Kaplan, J. T. (2017). Decoding the neural representation of story meanings across languages. *Human Brain Mapping*, 38(12), 6096–6106.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186.

DiMaggio, P. (1982). Cultural Capital and School Success: The Impact of Status Culture Participation on the Grades of U.S. High School Students. *American Sociological Review*, 47(2), 189–201.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2), 179–211.

Fast, L. A., & Funder, D. C. (2008). Personality as manifest in word use: correlations with self-report, acquaintance report, and behavior. *Journal of Personality and Social Psychology*, 94(2), 334–346.

Firth, R. J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), 378–382.

Gage, P. (1994). A new algorithm for data compression. *C Users Journal*, 12(2), 23–38.

Garcia, D., & Sikström, S. (2014). The dark side of Facebook: Semantic representations of status updates predict the Dark Triad of personality. *Personality and Individual Differences*, 67, 92–96.

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16), E3635-E3644.

Garten, J., Hoover, J., Johnson, K. M., Boghrati, R., Iskiwitch, C., & Dehghani, M. (2018). Dictionaries and distributions: Combining expert knowledge and large scale textual data content analysis. *Behavior Research Methods*, 50(1), 344–361.

Garten, J., Kennedy, B., Sagae, K., & Dehghani, M. (2019). Measuring the importance of context when modeling language comprehension. *Behavior Research Methods*.

Geigle, C. (2016). Inference Methods for Latent Dirichlet Allocation. http://times.cs.uiuc.edu/course/598f16/notes/lda-survey.pdf

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. "O'Reilly Media, Inc."

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *The Journal of Artificial Intelligence Research*.

Goldberg, Y [@yoavgo]. (2020, December 13). (i realize people no longer do LDA, but:) Do you also feel that the topics produced by sampling-based implementations are consistently superior to those produced by variational solvers? and if so, how come the variational methods came to be the dominant implementations? [Tweet]. *Twitter*. https://twitter.com/yoavgo/status/1338120353260986369

Gonen, H., & Goldberg, Y. (2019, June). Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 609-614).

Gottschalk, L. A., & Gleser, G. C. (1979). *The Measurement of Psychological States Through the Content Analysis of Verbal Behavior*. University of California Press.

Graham, J., Haidt, J., & Nosek, B. A. (2009). Liberals and conservatives rely on different sets of moral foundations. *Journal of Personality and Social Psychology*, 96(5), 1029–1046.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005. (Vol. 4, pp. 2047-2052). IEEE.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl 1, 5228–5235.

Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H., & Eichstaedt, J. C. (2017). Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18, 43-49.

Haidt, J., & Joseph, C. (2004). Intuitive ethics: How innately prepared intuitions generate culturally variable virtues. *Daedalus*, 133(4), 55-66.

Harris, Z. S. (1954). Distributional Structure. *Word & World*, 10(2-3), 146–162.

Healy, W., & Bronner, A. (1925). Judge Baker Foundation Case Studies. Case II, 5a, 7a, and 17a.

Henrich, J., Heine, S. J., & Norenzayan, A. (2010). The weirdest people in the world? *The Behavioral and Brain Sciences*, 33(2-3), 61–83; discussion 83–135.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics: A Journal of Statistics for the Physical, Chemical, and Engineering Sciences*, 12(1), 55–67.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1), 177–196.

Hogenraad, R. (2003). The Words that Predict the Outbreak of Wars. *Empirical Studies of the Arts*, 21(1), 5–20.

Hong, L., & Davison, B. D. (2010, July). Empirical study of topic modeling in Twitter. In *Proceedings of the first workshop on social media analytics* (pp. 80-88).

Hoover, J., Atari, M., Mostafazadeh Davani, A., Kennedy, B., Portillo-Wightman, G., Yeh, L., Kogon, D., & Dehghani, M. (2019). Bound in Hatred: The role of group-based morality in acts of hate. https://doi.org/10.31234/osf.io/359me

Hoover, J., Portillo-Wightman, G., Yeh, L., Havaldar, S., Davani, A. M., Lin, Y., ... & Dehghani, M. (2020). Moral Foundations Twitter Corpus: A collection of 35k tweets annotated for moral sentiment. *Social Psychological and Personality Science*, 11(8), 1057-1071.

Hornik, K., Stinchcombe, M., White, H., & Others. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks: The Official Journal of the International Neural Network Society*, 2(5), 359–366.

Howard, J., & Ruder, S. (2018, July). Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328-339).

Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2020). Compositionality Decomposed: How do Neural Networks Generalise? *The Journal of Artificial Intelligence Research*, 67, 757–795.

Hussey, I., & Hughes, S. (2020). Hidden Invalidity Among 15 Commonly Used Measures in Social and Personality Psychology. *Advances in Methods and Practices in Psychological Science*, 3(2), 166–184.

Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 8, No. 1).

Iovet︠s︡-Tereshchenko, N. (1936). Friendship-love in adolescence. Allen & Unwin.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917–963.

Jagarlamudi, J., Daumé, H., III, & Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 204–213.

Jin, X., Wei, Z., Du, J., Xue, X., & Ren, X. (2019). Towards Hierarchical Importance Attribution: Explaining Compositional Semantics for Neural Sequence Models. In *Proceedings International Conference on Learning Representations*. https://openreview.net/forum?id=BkxRRkSKwr

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. *Machine Learning: ECML-98*, 137–142.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), 11–21.

Jordan, K. N., Sterling, J., Pennebaker, J. W., & Boyd, R. L. (2019). Examining long-term trends in politics and culture through language of political leaders and cultural institutions. *Proceedings of the National Academy of Sciences*, 116(9), 3476–3481.

Joulin, A., Grave, É., Bojanowski, P., & Mikolov, T. (2017, April). Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (pp. 427-431).

Kahn, J. H., Tobin, R. M., Massey, A. E., & Anderson, J. A. (2007). Measuring emotional expression with the Linguistic Inquiry and Word Count. *The American Journal of Psychology*, 120(2), 263–286.

Kanze, D., Conley, M. A., & Higgins, E. T. (2019). The motivation of mission statements: How regulatory mode influences workplace discrimination. *Organizational Behavior and Human Decision Processes*.

Kennedy, B., Atari, M., Mostafazadeh Davani, A., Hoover, J., Omrani, A., Graham, J., & Dehghani, M. (2020). Moral Concerns are Differentially Observable in Language. PsyArxiv. https://doi.org/10.31234/osf.io/uqmty

Kennedy, B., Atari, M., Mostafazadeh Davani, A., Yeh, L., Omrani, A., Kim, Y., Coombs, K., Jr., Havaldar, S., Portillo-Wightman, G., Gonzalez, E., Hoover, J., Azatian, A., Hussain, A., Lara, Austin, Olmos, G., Omary, A., Park, C., Wang, C., … Dehghani, M. (2020). The Gab Hate Corpus: A collection of 27k posts annotated for hate speech. PsyArXiv, 18. https://doi.org/10.31234/osf.io/hqjxn

Kennedy, B., Jin, X., Davani, A. M., Dehghani, M., & Ren, X. (2020). Contextualizing Hate Speech Classifiers with Post-hoc Explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5435-5442).

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., & Fidler, S. (2015, December). Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2* (pp. 3294-3302).

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences of the United States of America*, 110(15), 5802–5805.

Kozlowski, A. C., Taddy, M., & Evans, J. A. (2019). The Geometry of Culture: Analyzing the Meanings of Class through Word Embeddings. *American Sociological Review*, 84(5), 905–949.

Krippendorff, K. (2004). Measuring the Reliability of Qualitative Text Analysis Data. *Quality and Quantity*, 38(6), 787–800.

Kučera, H., & Francis, W. N. (1970). Computational analysis of present-day American English. Brown University Press.

Kurenkov, A. (2020). Lessons from the PULSE Model and Discussion [Review of Lessons from the PULSE Model and Discussion]. *The Gradient.* https://thegradient.pub/pulse-lessons/

Laver, M., Benoit, K., & Garry, J. (2003). Extracting Policy Positions from Political Texts Using Words as Data. *The American Political Science Review*, 97(2), 311–331.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

LeCun, Y., Touresky, D., Hinton, G., & Sejnowski, T. (1988, June). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school* (Vol. 1, pp. 21-28).

Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Machine Learning*, 1188–1196.

Lewis, M., & Lupyan, G. (2020). Gender stereotypes are reflected in the distributional structure of 25 languages. *Nature Human Behaviour*, 4(10), 1021–1028.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1–167.

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 63–70.

Louwerse, M. M., McCarthy, P. M., McNamara, D. S., & Graesser, A. C. (2004). Variation in language and cohesion across written and spoken registers. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 26, No. 26).

Maks, I., Izquierdo, R., Frontini, F., Agerri, R., Azpeitia, A., & Vossen, P. (2014). Generating Polarity Lexicons with WordNet propagation in five languages. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 1155–1161.

Martindale, C. (2007). Creativity, primordial cognition, and personality. *Personality and Individual Differences*, 43(7), 1777–1785.

Martin, J. H., & Jurafsky, D. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson/Prentice Hall Upper Saddle River.

Matsuo, A., Sasahara, K., Taguchi, Y., & Karasawa, M. (2019). Development and validation of the Japanese Moral Foundations Dictionary. *PloS One*, 14(3), e0213343.

Mauss, I. B., & Robinson, M. D. (2009). Measures of emotion: A review. *Cognition and Emotion*, 23(2), 209–237.

McCormick, C. (2019, May 14). BERT Word Embeddings Tutorial. https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/

Medin, D., Bennis, W., & Chandler, M. (2010). Culture and the Home-Field Disadvantage. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, 5(6), 708–713.

Mehl, M. R., & Pennebaker, J. W. (2003). The sounds of social life: a psychometric analysis of students' daily social environments and natural conversations. *Journal of Personality and Social Psychology*, 84(4), 857–870.

Mehl, M. R., Pennebaker, J. W., Crow, D. M., Dabbs, J., & Price, J. H. (2001). The Electronically Activated Recorder (EAR): A device for sampling naturalistic daily activities and conversations. *Behavior research methods, instruments, & computers*, 33(4), 517-523.

Mehl, M. R., Robbins, M. L., & Holleran, S. E. (2012). How taking a word for a word can be problematic: Context-dependent linguistic markers of extraversion and neuroticism. *Journal of Methods and Measurement in the Social Sciences*, 3(2), 30–50.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635.*

Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013). Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 889–892.

Mergenthaler, E. (1996). Emotion–abstraction patterns in verbatim protocols: A new way of describing psychotherapeutic processes. *Journal of Consulting and Clinical Psychology*, 64(6), 1306–1315.

Meyer, D., Hornik, K., & Feinerer, I. (2008). Text Mining Infrastructure in R. *Journal of Statistical Software*, 25(5), 1–54.

Meyer, M. L., Hershfield, H. E., Waytz, A. G., Mildner, J. N., & Tamir, D. I. (2019). Creative expertise is associated with transcending the here and now. *Journal of Personality and Social Psychology*, 116(4), 483–494.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Google Books Team, Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., & Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014), 176–182.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *The 11th Annual cConference of the International Speech Communication Association*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013, December). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2* (pp. 3111-3119).

Miller, G. A. (1998). WordNet: An Electronic Lexical Database. MIT Press.

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing Semantic Coherence in Topic Models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 262–272.

Mitchell, M., Hollingshead, K., & Coppersmith, G. (2015). Quantifying the language of schizophrenia in social media. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, 11–20.

Mooijman, M., Hoover, J., Lin, Y., Ji, H., & Dehghani, M. (2018). Moralization in social networks and the emergence of violence during protests. *Nature Human Behaviour*, 2(6), 389–396.

Murdoch, W. J., Liu, P. J., & Yu, B. (2018, February). Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *International Conference on Learning Representations*.

Newman, M. L., Groom, C. J., Handelman, L. D., & Pennebaker, J. W. (2008). Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3), 211–236.

Opoku, R. A., Hultman, M., & Saheli-Sangari, E. (2008). Positioning in market space: The evaluation of Swedish universities' online brand personalities. *Journal of Marketing for Higher Education*, 18(1), 124–144.

Park, G., Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Kosinski, M., Stillwell, D. J., Ungar, L. H., & Seligman, M. E. P. (2015). Automatic personality assessment through social media language. *Journal of Personality and Social Psychology*, 108(6), 934–952.

Partee, B. (1995). Lexical semantics and compositionality. *An Invitation to Cognitive Science: Language*, 1, 311–360.

Paullada, A., Raji, I. D., Bender, E., Denton, E., & Hanna, A. (2020). Data and its (dis)contents: A survey of dataset development and use in machine learning research. In *arXiv [cs.LG]*. *arXiv*. http://arxiv.org/abs/2012.05345

Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). LIWC2007: Linguistic inquiry and word count. Austin, Texas: Liwc. Net.

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). The development and psychometric properties of LIWC2015.

Pennebaker, J. W., Chung, C. K., Frazee, J., Lavergne, G. M., & Beaver, D. I. (2014). When small words foretell academic success: the case of college admissions essays. *PloS One*, 9(12), e115844.

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6), 1296–1312.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers) (pp. 2227-2237).

Pietraszkiewicz, A., Formanowicz, M., Gustafsson Sendén, M., Boyd, R. L., Sikström, S., & Sczesny, S. (2019). The big two dictionaries: Capturing agency and communion in natural language. *European Journal of Social Psychology*, 49(5), 871–887.

Powers, D. M. (1998). Applications and explanations of Zipf's law. In *New methods in language processing and computational natural language learning*.

Pulverman, C. S., Boyd, R. L., Stanton, A. M., & Meston, C. M. (2017). Changes in the sexual self-schema of women with a history of childhood sexual abuse following expressive writing treatment. *Psychological Trauma: Theory, Research, Practice and Policy*, 9(2), 181–188.

Pury, C. L. (2011). Automation can lead to confounds in text analysis: Back, Küfner, and Egloff (2010) and the not-so-angry Americans. *Psychological science*, 22(6), 835.

Pyszczynski, T., & Greenberg, J. (1987). Self-regulatory perseveration and the depressive self-focusing style: a self-awareness theory of reactive depression. *Psychological Bulletin*, 102(1), 122–138.

Ratner, K., Burrow, A. L., Burd, K. A., & Hill, P. L. (2019). On the conflation of purpose and meaning in life: A qualitative study of high school and college student conceptions. *Applied Developmental Science*, 1–21.

Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G.-Z. (2017). Deep Learning for Health Informatics. *IEEE Journal of Biomedical and Health Informatics*, 21(1), 4–21.

Řehůřek, R., & Sojka, P. (2011). Gensim—statistical semantics in python. Retrieved from *gensim.org*.

Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning* (ICML-03), 616–623.

Resnik, P., & Hardisty, E. (2010). *Gibbs sampling for the uninitiated*. Maryland Univ College Park Inst for Advanced Computer Studies.

Richie, R., Zou, W., & Bhatia, S. (2019). Predicting high-level human judgment across diverse behavioral domains. *Collabra: Psychology*, 5(1).

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5), 503–520.

Robinson, R. L., Navea, R., & Ickes, W. (2013). Predicting final course performance from students' written self-introductions: A LIWC analysis. *Journal of Language and Social Psychology*, 32(4), 469–479.

Rochon, E., Saffran, E. M., Berndt, R. S., & Schwartz, M. F. (2000). Quantitative analysis of aphasic sentence production: further development and new data. *Brain and Language*, 72(3), 193–218.

Rodríguez-Arauz, G., Ramírez-Esparza, N., Pérez-Brena, N., & Boyd, R. L. (2017). Hablo Inglés y Español: Cultural self-schemas as a function of language. *Frontiers in Psychology*, 8, 885.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.

Rouhizadeh, M., Ungar, L., Buffone, A., & Schwartz, H. A. (2016). Using syntactic and semantic context to explore psychodemographic differences in self-reference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2054–2059.

Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, 15–18.

Rude, S., Gortner, E.-M., & Pennebaker, J. W. (2004). Language use of depressed and depression-vulnerable college students. *Cognition and Emotion*, 18(8), 1121–1133.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.

Sagi, E., & Dehghani, M. (2014). Measuring Moral Rhetoric in Text. *Social Science Computer Review*, 32(2), 132–144.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, & Sung Hyon Myaeng. (2006). Some Effective Techniques for Naive Bayes Text Classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11), 1457–1466.

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2019). The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1668–1678.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks: The Official Journal of the International Neural Network Society*, 61, 85–117.

Schwartz, H., Eichstaedt, J., Kern, M., Dziurzynski, L., Lucas, R., Agrawal, M., ... & Ungar, L. (2013). Characterizing geographic variation in well-being using tweets. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 7, No. 1).

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E. P., & Ungar, L. H. (2013).

Personality, gender, and age in the language of social media: the open-vocabulary approach. *PloS One*, 8(9), e73791.

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54–63.

Selling, L. S. (1932). The autobiography as a psychiatric technique. *The American Journal of Orthopsychiatry*, 2(2), 162.

Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1715-1725).

Singh, C., Murdoch, W. J., & Yu, B. (2018, September). Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*.

Stirman, S. W., & Pennebaker, J. W. (2001). Word Use in the Poetry of Suicidal and Nonsuicidal Poets. *Psychosomatic Medicine*, 63(4), 517.

Stone, P. J., Dunphy, D. C., & Smith, M. S. (1966). The general inquirer: A computer approach to content analysis. 651.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645-3650).

Sumner, C., Byers, A., & Shearing, M. (2011). Determining personality traits & privacy concerns from facebook activity. *Black Hat Briefings*, 11(7), 197–221.

Sumpter, R. S. (2001). News about news: John G. Speed and the first newspaper content analysis. *Journalism History*, 27(2), 64–72.

Tackman, A. M., Sbarra, D. A., Carey, A. L., Donnellan, M. B., Horn, A. B., Holtzman, N. S., Edwards, T. S., Pennebaker, J. W., & Mehl, M. R. (2019). Depression, negative emotionality, and self-referential language: A multi-lab, multi-measure, and multi-language-task research synthesis. *Journal of Personality and Social Psychology*, 116(5), 817–834.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1), 24–54.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2005). Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems* 17 (pp. 1385–1392). MIT Press.

Teh, Y. W., Newman, D., & Welling, M. (2007). A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems* 19 (pp. 1353–1360). MIT Press.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593–4601.

Tetlock, P. E. (1981). Personality and isolationism: Content analysis of senatorial speeches. *Journal of Personality and Social Psychology*, 41(4), 737–743.

Tetlock, P. E. (1983). Cognitive style and political ideology. *Journal of Personality and Social Psychology*, 45(1), 118–126.

Thomas, C. B., & Duszynski, K. R. (1985). Are words of the Rorschach predictors of disease and death? The case of“ whirling.” *Psychosomatic Medicine*.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 58(1), 267–288.

Torrey, L., & Shavlik, J. (2010). Transfer Learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (pp. 242–264). IGI Global.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. U., & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). Curran Associates, Inc.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018, 7068349.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of Personality and Social Psychology*, 54(6), 1063–1070.

Wetzel, L. (2006). Types and tokens. https://stanford.library.sydney.edu.au/entries/types-tokens/

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.

Yaden, D. B., Eichstaedt, J. C., Kern, M. L., Smith, L. K., Buffone, A., Stillwell, D. J., Kosinski, M., Ungar, L. H., Seligman, M. E. P., & Schwartz, H. A. (2018). The Language of Religious Affiliation: Social, Emotional, and Cognitive Differences. *Social Psychological and Personality Science*, 9(4), 444–452.

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, 1445–1456.

Young, L., & Soroka, S. (2012). Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 29(2), 205–231.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*, 13(3), 55–75.

Zawadzki, B., & Lazarsfeld, P. (1935). The psychological consequences of unemployment. *The Journal of Social Psychology*, 6(2), 224-251.