Behav Res (2017) 49:538–547
DOI 10.3758/s13428-016-0722-4

REVIEW

TACIT: An open-source text analysis, crawling, and interpretation tool

Morteza Dehghani[1] · Kate M. Johnson[1] · Justin Garten[1] · Reihane Boghrati[1] · Joe Hoover[1] · Vijayan Balasubramanian[1] · Anurag Singh[1] · Yuvarani Shankar[1] · Linda Pulickal[1] · Aswin Rajkumar[1] · Niki Jitendra Parmar[1]

Published online: 4 March 2016

Abstract As human activity and interaction increasingly take place online, the digital residues of these activities provide a valuable window into a range of psychological and social processes. A great deal of progress has been made toward utilizing these opportunities; however, the complexity of managing and analyzing the quantities of data currently available has limited both the types of analysis used and the number of researchers able to make use of these data. Although fields such as computer science have developed a range of techniques and methods for handling these difficulties, making use of those tools has often required specialized knowledge and programming experience. The Text Analysis, Crawling, and Interpretation Tool (TACIT) is designed to bridge this gap by providing an intuitive tool and interface for making use of state-of-the-art methods in text analysis and large-scale data management. Furthermore, TACIT is implemented as an open, extensible, plugin-driven architecture, which will allow other researchers to extend and expand these capabilities as new methods become available.

Keywords Automated text analysis · Language-based assessment · Computational linguistics · Corpus analysis tools

Individuals' daily activities increasingly take place in the digital space. The average adult spends 6.15 h/day online, more than a quarter of which they spend on social networking (Mander, 2015). As of 2015, 74 % of online adults use social media sites such as Facebook or Twitter (Mander, 2015),2 and many regularly turn to these sites for everything from social support to daily news information (Duggan, Ellison, Lampe, Lenhart, & Madden, 2015; T. M. I. Project, 2015).

This rise of digital engagement generates a large quantity of wide-ranging information about naturally occurring social interactions, and much of this information is in the form of written language. Computer-assisted content analysis techniques have been developed to help assess these kinds of data and to automate parts of the research process that were previously infeasible, due to high resource cost or small datasets.

The first programs for analyzing text for social scientific purposes were developed as early as the 1960s (General Inquirer; Stone, Dunphy, & Smith, 1966). These word count programs used researcher-generated dictionaries (i.e., lists of words that represent a category) and automatically counted the number of times the words in that list occurred in a given text. Building off of these algorithms, user-friendly programs such as Diction (Hart, 1984) and Linguistic Inquiry Word Count (LIWC; Tausczik & Pennebaker, 2010) were developed to provide researchers the ability to automatically process text using their own dictionaries and to calculate the percentage of the text represented by each category.

The proprietary program LIWC was specifically developed to assess text using validated, psychologically meaningful categories, and it has been one of the most widely adopted language analysis programs within the field. Language analysis research using LIWC has provided valuable insight into psychological processes and behavioral outcomes across a variety of content domains, including the stability of relationships

Electronic supplementary material The online version of this article (doi:10.3758/s13428-016-0722-4) contains supplementary material, which is available to authorized users.

✉ Morteza Dehghani
mdehghan@usc.edu

[1] University of Southern California, 3620 S. McClintock Avenue
Los Angeles, CA, USA

(Ireland et al., 2011), whether or not the individual is in pain (Rude, Gortner, & Pennebaker, 2004), and honesty (Newman, Pennebaker, Berry, & Richards, 2003), among many other important findings.

Other single-purpose programs have been developed to address an array of additional linguistic analysis goals that do consider syntactic structure. For example, specialized programs are available to assess the readability (and cohesion) of a given piece of text (Coh-Metrix; McNamara et al., 2014), the syntactic complexity of a document (Syntactic Complexity Analyzer; Lu, 2010), and a document's level of lexical sophistication (TAALES; Kyle & Crossley, 2015).

Although each of these programs provides important tools for assessing specific components of text, the static, single-purpose nature of these and other available language analysis applications has been a consistent barrier to using state-of-the-art techniques for conducting text-based research in psychology. Accordingly, researchers have begun using a range of new algorithms that allow for more sophisticated and robust text analysis. Although these algorithms vary considerably in both their goals and their mathematical structures, they collectively have enabled researchers to analyze a variety of language components that cannot be captured by word count methods alone.

However, widespread use of many of these algorithms, developed mainly by computer scientists and computational linguists, has been hampered by two major barriers. First, the considerable technical expertise required to implement these algorithms in the existing software can be prohibitive for researchers without backgrounds in computer science or related fields. For example, many packages such as *tm* have been developed to conduct natural-language processing algorithms in the free data-analysis program R (Meyer et al., 2015), but the steep learning curve needed to familiarize one's self with R's code-based interface has limited its adoption within the field, despite its expansive capabilities. Furthermore, the constant increase of computational processing power and the proliferation of new algorithms makes it difficult for researchers to maintain working knowledge of state-of-the-art methods.

Second, most of the existing user-friendly NLP programs (and packages), such as RapidMiner (Akthar & Hahne, 2012), SAS Text Miner (Abell, 2014), or SPSS Modeler (IBM Corp., 2011), charge either a large software fee up front or a subscription fee. The cost of these programs can be prohibitively expensive for junior researchers and researchers looking to integrate new techniques into their research toolbox.

Accordingly, it is important that we develop tools that can be dynamically adapted with the changing field, to leverage these advances as well as make sophisticated analytical procedures more accessible for researchers. In this article, we introduce TACIT: Text Analysis, Crawling and Investigation Tool. TACIT is an open-source architecture that establishes a pipeline between the various stages of text-based research by integrating tools for text mining, data cleaning, and analysis under a single user-friendly architecture.[1] In addition to being prepackaged with a range of easily applied, cutting-edge methods, TACIT's design also allows other researchers to write their own plugins.

TACIT

As we discussed above, though several limited-method tools for text analysis are already available (e.g., LIWC), and some have become part of standard statistical packages (e.g., SPSS Text Analytics), a unified, open-source architecture for gathering, managing, and analyzing text does not exist. The Computational Social Science Laboratory (CSSL) team at University of Southern California developed TACIT to address some of these shortcomings. TACIT's plugin architecture has three main components: (1) crawling plugins, for automated text collection from online sources; (2) corpus management, for applying standard text preprocessing in order to prepare and store corpora; and (3) analysis plugins, including LIWC-type[2] word count, topic modeling, sentiment analysis, clustering, and classification. TACIT's open-source plugin platform allows the architecture to easily adapt with the rapid developments in automated text analysis. Certain plugins in TACIT provide a researcher-friendly interface to well-known libraries used in computational linguistics (e.g., Stanford CoreNLP; Manning et al., 2014) and machine learning (e.g., Weka 3; Bouckaert et al., 2010), while other algorithms have been implemented specifically for TACIT (e.g., the crawlers). Given that computational text analysis is slowly becoming a field norm, TACIT can vastly increase psychologists' access to both large-scale textual data and cutting-edge text analysis methods. Below, we discuss each of TACIT's components in detail, using the output of one of the crawlers as an example dataset analyzed by different plugins.

TACIT crawlers

The availability of vast amounts of human-related data via digital content has been the main impetus for the growing importance of text analysis in psychology, but accessing and processing this data into an analyzable format can be quite daunting. We designed TACIT's crawler plugins to automate this data compilation stage. The six built-in TACIT crawlers search and download relevant content from various historical (the Latin Library Crawler), political (the US Supreme Court and US Congress Crawlers), and social media (the Reddit, Twitter, and Stack Exchange Crawlers) services to make these

[1] TACIT can be downloaded from http://tacit.usc.edu.

[2] All the comparisons to LIWC are based on the 2010 version of the software.

data available for further processing in TACIT or any other text-processing tool. All data that are downloaded with TACIT's crawlers are automatically converted by the Corpus Management tool into a corpus (annotated collection of texts), which can be used for analysis within the TACIT program or exported as plain text files for use in other analysis software. Researchers can also use the existing crawling architecture in TACIT to develop their own crawlers and easily incorporate the new crawler in Corpus Management. Below we describe each of the built-in crawlers in detail.

US congress crawler Transcripts of the speeches given at the US Congress have been used to assess a wide variety of social and political psychological research questions, including the link between congressional prosociality and approval ratings (Frimer, Aquino, Gebauer, Zhu, & Oakes, 2015), differences in partisan emotional states (Wojcik, Hovasapian, Graham, Motyl, & Ditto, 2015), and partisan language use differences (Yu, Kaufmann, & Diermeier, 2008). The US Congress Crawler collects speech transcription data from the Library of Congress THOMAS website (http://thomas.loc.gov/home/thomas.php) for present-day speeches to as far back as the 101st Congress. Given the Congress number, senator/representative details, and other filtering options, the US Congress Crawler automatically scans and collects speech transcription data and writes that data into text files for analysis. This crawler also saves additional information about the speeches in a summary CSV file (e.g., the name of the speaker, the date of the speech).

We used the Senate Crawler to compile a dataset of transcripts of Senate speeches from 9/10/96 to 9/10/01, and another for 9/12/01 to 9/12/06. We only collected five random transcripts from each senator. This corpus will be used to guide the reader throughout the article. Please see the supplementary materials for details about how this crawl and the analyses discussed in the coming sections were performed.

Reddit crawler Reddit is the 10th most visited site in the US and the 31st in the world (www.alexa.com/siteinfo/reddit.com), and the discussion structure of Reddit, plus its up/down vote mechanism, provides a rich platform for psychological research (Van Mieghem, 2011), though few projects have made use of these data to date. Given the open nature of Reddit and the wide variety of topics represented on the site, we believe this underutilization is likely due to the difficulty of accessing and parsing Reddit data. The Reddit Crawler provides access to the data available on Reddit.com. Through its interface to jReddit (https://github.com/jReddit/jReddit), this crawler provides various filtering options that can be used to search for specific topics, keywords and authors, from subreddits through either the full website, the trending data, or the top/controversial section of the website. These data, including vote counts, are then saved in a format that is accessible and analyzable by the TACIT tool as well as by other language analysis algorithms.

Twitter crawler Twitter (www.twitter.com) is arguably the most popular site used by psychologists for the analysis of various social phenomena on social media, likely due to the fact that Twitter provides a free, rather friendly application programming interface (API) for accessing tweet text and user social connections information up to 48 h old. This information can then be used to investigate community-related phenomena (e.g., belief propagation, homophily). Recently, several publications have increased the excitement about the use of Twitter data for social and personality research, due to its ability to predict important psychological outcomes. Some examples of these outcomes include depression (De Choudhury, Gamon, Counts, & Horvitz, 2013), county-level heart disease mortality (Eichstaedt et al., 2015), and purity homophily (Dehghani et al., 2016).

Using twitter4j (http://twitter4j.org/), the TACIT Twitter Crawler collects tweets and relevant information about each tweet (such as number of favorites, retweets, geolocation, etc.) from the Twitter public stream. These data are then saved into a file that can be queried using various information about the tweets, and subsets of the data can be written in plain-text format for further analysis. Due to the nature of Twitter's API system, users must have a Twitter account and need to go through a brief authentication process on Twitter before being able to use this crawler.

Supreme court crawler TACIT provides access to transcriptions and audio files (where available) from the Oyez Project at ITT Chicago–Ken College of Law's Supreme Court speech archive (www.oyez.org/cases). Users can access cases ranging from the 19th century to the present day. The archive can be searched by Supreme Court number or by the topic of the case (e.g., voting rights, double jeopardy).

Latin library crawler The Latin Library Crawler enables users to download text available at the Latin Library (www.thelatinlibrary.com/), a collection of public-domain Latin texts. We believe this crawler can be a valuable tool for computational classicists interested in using modern text-processing techniques.

Stack exchange crawler Stack Exchange (http://stackexchange.com/) is a popular platform of over 130 Q&A communities, with millions of users. The TACIT Stack Exchange Crawler can be used to gather text from any of the communities, on the basis of tags, users, and questions. The crawled data include the question text, the answers, user information, and up/down votes for the answers.

Corpus management

Corpora can represent a range of data, including the works of a single author, a collection of tweets on a single day that used a specific hashtag, or a collection of news articles from a particular source. The text can be raw or can be annotated to contain additional information, such as part of speech tags (e.g., Brown Corpus; Leech, Garside, & Atwell, 1983) or tags denoting whether a bunch of text is spam (e.g., The Enron Spam dataset; Klimt & Yang, 2004).

Any toolkit that aims to provide language analysis must provide an easy and efficient way to manage different types of corpora, and TACIT's Corpus Management tool aims to provide a simple user interface to accomplish this task. Corpus Management allows users' corpora to be compiled in TACIT at the beginning of a project so that the process of loading and managing the data need not to be repeated multiple times. The Corpus Management tool can import user data in multiple formats, such as JSON, Microsoft Word, and CSV files. After creating a corpus in TACIT, the program will automatically record and provide a summary log of how and when the data were collected, as well as all analyses performed on that corpus (e.g., type of analysis, date and time of analysis). When such corpora are used for any analysis, users have the ability to filter their data on the basis of various parameters, including tweet/comment data, keywords, and associated hashtags.

Preprocessing

After data have been imported to TACIT (through either crawling or direct import), an array of preprocessing features are available to clean the data and prepare them for further analysis. Preprocessing textual data generally helps improve the accuracy of analyses by reducing the vocabulary size. Users can use TACIT's preprocessing features to remove stopwords (common words such as "a" or "the" that in most circumstances provide no useful information), perform automatic stemming (i.e., map each word to its root; e.g., swam, swim, swimmer, swimming → swim), and standardize text (e.g., convert all text to lowercase or remove extraneous information). TACIT automatically detects the language of the text and uses an implementation of the Porter Stemmer (Porter, 2006) or Snowball stemming (Porter, 2001) to apply the correct stemming rules for that language. It is important to note that that this feature is different from LIWC-style stemming (i.e., the use of "*" in dictionaries), because LIWC matches string patterns (swim, swimmer, swimming → swim*) and does not necessary map words to their roots (e.g., "swam" does not get mapped to "swim*" in LIWC). We plan to include an option for automatic spelling correction as part of the preprocessing toolkit.

Word count

Word counting is a simple and widely used approach for analyzing text that counts the number of words or phrases in a given text and categorizes them on the basis of previously defined user-generated categories. Arguably, the most popular natural-language platform in social psychology is LIWC, developed by James Pennebaker and his collaborators (Pennebaker, Booth, & Francis, 2007; Tausczik & Pennebaker, 2010). LIWC has been used in a variety of different tasks, including demonstrating relationships between the use of first person singular pronouns and negative experiences and depression (Rude et al., 2004; Stirman & Pennebaker, 2001); between extraversion and the use of shorter words and less complex language, but longer written passages (Mehl, Gosling, & Pennebaker, 2006; Pennebaker & King, 1999); and between deception and the use of words relating to motion (Newman et al., 2003).

TACIT provides two plugins that perform word counts: LIWC and Standard Word Count. The LIWC-style word count plugin was developed using the LIWC documentation (Pennebaker, Francis, & Booth, 2001) and by running the program on different types of corpora to understand and implement the algorithm. We compared the results of our implementation and the original LIWC software on a variety of test texts, including files from Project Gutenberg (https://www.gutenberg.org/). Except for very rare cases (e.g., some occurrences of double hyphens), our plugin yielded results that were exact matches to LIWC's up to the hundredth decimal point. Although TACIT provides the word count capability of LIWC, users must import their own dictionaries, since our program does not provide any of the LIWC dictionary categories.

The Standard Word Count plugin was developed exclusively for the TACIT tool as a more comprehensive approach to word count techniques. Rather than relying only on static categories, the Standard Word Count tool uses Apache OpenNLP (http://opennlp.apache.org/) to automatically segment sentences, tokenize words, and find the part-of-speech tags of all words in the text. Analyses using this tool report the counts for the part-of-speech tags as well as the categories in the user's dictionaries. We believe that this is a more comprehensive approach to segmenting sentences and counting parts of speech than is LIWC's approach (e.g., locating "." as marker for sentences), because it automatically identifies and accounts for different uses of punctuation (e.g., "." used as a decimal place, or in an abbreviation, or "…").

Generally, the words in a category are all treated as being equally important, and thus are assigned the same weight during analysis (i.e., no weight). However, a weighted word count can allow the program to incorporate different weights for each word. In weighted word counts, higher weights are assigned to words that are particularly diagnostic of a category, resulting in higher output scores when that word is present, even if the word occurs less frequently than other category words. Although LIWC word count is generally unweighted, both of TACIT's word count plugins support weighted dictionaries.

To demonstrate the utility of word counts, we ran an analysis comparing the transcriptions of Congress speeches given prior to and after September 11, 2001 (hereafter, "9/11"). Previous research has shown that traumatic events often result in the use of language related to an increase in collective orientation (Cohn et al., 2004). Therefore, our hypothesis was that we would see an increase in the use of words related to social orientations and in the use of moral loyalty rhetoric. We used the LIWC word count plugin to investigate the frequency of words related to social processes (e.g., talk, share, friends) and the moral rhetoric of loyalty to ingroup members. The first analysis was done using the LIWC 2007 dictionary (Pennebaker et al., 2007), and the second using the Moral Foundations Dictionary (Graham, Haidt, & Nosek, 2009). Using repeated measures analysis of variance, we found that there were increases in the use of words related to social processes, $F(1, 3305) = 31.627$, $p < .001$, and to loyalty, $F(1, 3305) = 82.256$, $p < .001$.

Co-occurrence analysis

It is often useful as a preliminary but important step in data analysis to study the relationships and patterns between the variables of interest. Co-occurrence algorithms offer insights into the interconnection between terms or entities within any given text. Two words are said to "co-occur" with each other if both appear in a defined window size—that is, within a certain number of words of each other (e.g., in "Fiji apples are red," "Fiji" and "red" co-occur in a window size of 4). TACIT's co-occurrence plugin calculates both "word-to-word level" and "multiword level" co-occurrences. At the word-to-word level, the co-occurrence plugin computes the frequency at which each pair of words co-occur in a text or corpus. At the multiword level, the co-occurrence tool allows researchers to specify multiple words of interest (e.g., "fiji," "red," and "island"). The tool then calculates how frequently all subcombinations of these words occur together (are "neighbors" to each other) within a researcher-specified window size (or "neighborhood"; e.g., "Fiji" and "red" are neighbors within the four-word neighborhood of "Fiji apples are red"). TACIT then provides an output file with the frequencies of the different word combinations and their locations within the files (i.e., filename and line number). Inferences can be drawn easily through the output of the co-occurrence analysis to locate and verify the relationships between given entities. The overall matrix of word pair frequencies can be used to locate clusters or synonyms and to learn the overall connections between the words in a document.

To demonstrate the utility of co-occurrence analysis, we ran an analysis on the Senate dataset investigating the frequencies at which the words "iraq," "war," and "wmd," and "clinton," "lewinsky," and "affair" co-occurred with each other. We ran this analysis once for the pre-9/11 dataset and once for the post-9/11 transcripts. We expected that we would find that "clinton," "lewinsky," and "affair" co-occurred frequently pre-9/11, and that "iraq," "war," and "wmd" would cluster post-9/11. We set the window size to 10 (neighborhood size) and the threshold to 2 (meaning that at least two of the three words needed to co-occur in the neighborhood). We found that the "Clinton" set co-occurred 25 times in the pre-9/11 dataset, and did not co-occur in the post set. Also, as expected, the "Iraq" set co-occurred 41 times in the post, and only three times in the pre, set.

Classification

Machine-learning tasks are often aimed at automatically sorting data into predetermined groups of interest. For this kind of problem, a popular solution is to use a supervised classification algorithm or "classifier," such as naive Bayes (Lewis, 1998), random forests (Breiman, 2001), or support vector machines (SVMs; Cortes & Vapnik, 1995). For example, researchers have used classifiers to predict the gender of blog authors (Mukherjee & Liu, 2010), the political affiliation of blog authors (Dehghani, Sagae, Sachdeva, & Gratch, 2014), and the religious affiliation of Twitter users (Nguyen & Lim, 2014). Although the algorithms employed by classifiers vary widely, they all share a basic three-step framework. First, the researcher organizes and labels documents according to the groups or classes the researcher is interested in assessing, and then splits the dataset into training and testing subsets. The classifier uses the training subset of the data to determine features (i.e., words) that distinguish the groups from each other. The classifier then uses this information to test how accurately the trained algorithm can predict class memberships for the remaining labeled documents in the test subset. Finally, if the trained classifier is sufficiently accurate, the algorithm can be used to predict which class any remaining or new unlabeled documents belong in.

TACIT provides two classification plugins: Naive Bayes and SVM. The Naive Bayes plugin interfaces with the Machine Learning for Language Toolkit (MALLET; McCallum, 2002) and can be used to investigate the separability of multiple labeled classes and predict the classes of unlabeled data. After using the Naive Bayes plugin to label unclassified data, users can conduct further analyses on the automatically classified dataset. Like the Naive Bayes plugin, the SVM plugin can be used to investigate the general separability of classes. However, it can also be used to gain greater insight into the textual features that distinguish classes. More specifically, the SVM plugin provides output that lists the numerical weights for each feature. By exploring this output, users can identify the words that are most strongly associated with class membership. The SVM plugin interfaces libSVM (Chang & Lin, 2011) and the Apache Commons Math library (Math, 2014).

Both of TACIT's classification plugins require that the researcher preorganize texts by class into separate directories/ corpora. Once the data are organized by group, TACIT's classification plugins automatically conduct training and test runs on the data, determine predictive accuracy, and calculate relevant classification statistics. Furthermore, both plugins support *k*-fold cross-validation of accuracy rates.

We ran the SVM plugin on the collected Senate dataset to see whether we could automatically classify transcriptions of speeches given prior to 9/11 and those given post-9/11. Basically, we wanted to investigate whether the use of words between the two dates was sufficiently different that transcripts of speeches could be automatically classified on the basis of when they were given. The average accuracy of classification was 97.29, with 10-fold cross-validation. Investigating the features for a fold that reached an accuracy closest to the mean revealed that the reason for the high classification accuracy was that the words with the highest weight were "2002," "2003," "2005," "2006," and "2004" for one class, and "1998," "2001," "2000," "1996," and "1999" for the other class. In other words, SVM was able to almost perfectly classify the documents on the basis of the dates that were mentioned in the speeches. Although this finding might not be of particular theoretical interest, it highlights the importance of understanding the data features that drive a set of results. By using TACIT's SVM tool to conduct feature investigation, researchers can easily gain deeper insight into their data.

To demonstrate the utility of the Naive Bayes classifier, we collected another sample of transcripts of Senate speeches given between 10/13/06 and 10/14/11. We then used the Naive Bayes plugin to run a three-way classification on the three datasets, investigating whether the language used in the three periods would be significantly different. With 10-fold cross-validation, the average classification accuracy was 60.55 % (chance = 33.33 %).

Cluster analysis

In contrast to classification, cluster analysis techniques automatically sort texts into groups based on similarities in the text itself, allowing researchers to identify new ways of grouping texts on the basis of similarities that they have not predetermined. TACIT includes cluster analysis plugins that interface with two of the most widely used cluster analysis algorithms: *k*-means clustering and hierarchical clustering. The *k*-means clustering tool (MacQueen, 1967) aims to cluster texts into a user-specified number of clusters (or groups), such that the texts included in each cluster are the nearest to the cluster's centroid (the prototypical document of that cluster) and have the farthest distance from other clusters' centroids. Once this optimization is performed, TACIT outputs a CSV file containing the membership information for the documents in the corpus.

Hierarchical clustering (Johnson, 1967), by comparison, does not require a prespecified number of clusters. Instead, this technique identifies the optimal number of clusters by starting with the assumption that all data points (in this case, documents) belong to one cluster. The algorithm then splits this root cluster into smaller child clusters based on the degree of similarity between the documents. These child clusters are recursively divided further until only singleton clusters remain. Besides providing a CSV file with membership information, TACIT also provides the option to visualize the clusters in a dendrogram that presents an overview of the hierarchies of clusters. Both of TACITs current clustering analysis plugins use Weka 3 (Bouckaert et al., 2010; Frank, Hall, & Trigg, 2006), which is a popular data-mining library in Java.

We used the *k*-Means plugin to perform an unsupervised classification of the Senate dataset. We included the pre- and post-9/11 transcripts and set *k* to 2, to find two clusters in the dataset. Given that we have ground truth for the transcripts (i.e., know the date on which each speech was given), we were able to check the accuracy of the *k*-means on this dataset. Manual verification of the documents in each cluster revealed that the accuracy of *k*-means was only 59 %. Even though this accuracy is significantly lower than SVM's result, it should be noted that, as compared to SVM, *k*-means is an unsupervised method and does not have access to the additional information about group membership that is provided to SVM during its training stage. Accordingly, in situations in which ground-truth information is limited or nonexistent, cluster analysis can provide valuable insight into the data patterns.

Topic modeling

While reading a document, we have an intuitive understanding that it focuses on particular topics. For example, at various levels of generality, by reading a news article we might find that the document is primarily about economics, sports, entertainment, the Trans-Pacific Partnership, or Wimbledon. Furthermore, most documents are not about a single topic, but instead are blends of multiple topics and attitudes. That blend can provide an effective signature for the document, providing a researcher with a way to compare multiple documents for similarity across a range of dimensions. Topic-modeling techniques (Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2004; Papadimitriou, Tamaki, Raghavan, & Vempala, 1998) are used to discover and describe the underlying topics within documents and changes in these topics over time. These techniques also help researchers explore how frequently different topics occur together and how words within a topic are associated. Psychological researchers have also used these techniques to identify how underlying topics within text associate with other external measures. For example, Eichstaedt and colleagues (Eichstaedt et al., 2015) analyzed Twitter tweets using the Latent Dirichlet Allocation (LDA) algorithm and found that higher use of words belonging to the topics hostility, interpersonal tension, and boredom/fatigue predicted county-level heart disease mortality with the same accuracy as all typical demographic (e.g., race, socioeconomic status) and physical (e.g., diabetes, smoking) predictors combined.

TACIT's LDA and *z*-label LDA topic modeling plugins were developed to accomplish this goal. LDA (Blei et al., 2003) starts from the assumption that documents are mixtures of multiple topics, and it aims to discover both the structure and the mix of topics for each document. TACIT's LDA plugin interfaces with MALLET and allows you to explore the structure of topics within a set of documents and the relations between them. Researchers can adjust the number of topics generated in order to focus the level of granularity, with smaller numbers of topics generating a few overarching themes, and larger numbers providing many tightly focused topics. Each topic is represented as a distribution over the words in the vocabulary, which means that every word in a corpus of documents is assigned a probability of occurrence for each topic. Accordingly, the gist of a given topic can be understood by exploring the words that are most likely to occur within that topic. This allows a researcher to see which words are most associated with topics and explore connections between them across the entire corpus. Similarly, each document is represented as a distribution over topics, such that each document is constituted by a mixture of probabilistically occurring topics, which provides a way to compare topic use across documents. TACIT's Seeded LDA plugin expands upon the capabilities of LDA by implementing the *z*-label LDA (Andrzejewski & Zhu, 2009) algorithm. *Z*-label LDA allows researchers to provide a word or list of words they are interested in exploring, called *seed words*. The algorithm then uses these seed words as the core of the topics that it generates, building the rest of the words around those key concepts.

We ran the LDA plugin to investigate the differences in the topics that emerge in the pre- and post-9/11 transcripts. When we set the number of topics to 50, the LDA results demonstrated that, unlike in the pre-9/11 dataset, several topics related to war, Afghanistan, Iraq, and Katrina appeared in the post dataset. Similarly, we ran *z*-label LDA, with the following sets of seed words: (1) "affair," "lewinsky," "clinton"; 2) "iraq," "afghanistan," "war"; (3) "tax," "money," "dollar." We expected that the third topic would form clearly in both the pre and post datasets, whereas the first would only form in the pre, and the second only in the post dataset. Consequently, we expected that words related to the seed words would form around the corresponding topics. As expected, in the pre dataset, in addition to the first set of seed words, other words, such as "warner," "gorton," "assist," and "secretary" for the first topic, and "income," "employ," "health," and "assist" for the third topic, gathered around the seeds. However, "iraq" and "afghanistan" did not appear in the second topic. For the post dataset, we got words such as "military," "saddam," "freedom," and "terrorist" in the post dataset for the second set of seed words, and "budget," "million," "income," and "economy" for the third set. In this dataset, the first set of seed words did not form a coherent topic.

Sentiment analysis

Sentiment analysis, also called *opinion mining*, is the task of determining latent subjective information about a given piece of text. Although generally this refers to positive or negative sentiments (e.g., words that predict positive or negative reviews), this technique is closely related to classification, and thus can also be applied to other latent variables of interest. The most basic methods determine a document's sentiment by using word count techniques (e.g., counting how often the words from each sentiment's predefined dictionary list occur). There are a number of challenges with using such methods, including issues such as the necessary incompleteness of lists and the fact that negation can flip the polarity of terms. However, recent work that has made use of distributed representations (Hirschberg & Manning, 2015) and syntactic parsing (Yang, Zhuang, & Zong, 2015) has demonstrated potential ways past some of these difficulties.

We are currently expanding TACIT to provide an array of methods to perform sentiment analysis on documents using user-defined sentiments. Through a simple interface, users will be able to perform sentiment analysis using tools including word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), doc2vec (Le & Mikolov, 2014), or algorithms such as Latent Semantic Analysis (Deerwester, Dumais, Landauer, & Furnas, 1990). To use these tools in TACIT, the user would need to provide some seed words for the category to be investigated (e.g., for Harm: "harm," "kill," "hurt"). Depending on the chosen technique, TACIT will calculate vector representations of these words and calculate the distance of the target document from these vectors. Apart from this, we are working to include a sentiment analysis plugin geared specifically toward Twitter data, to tackle Twitter-specific abbreviations and slang used in social media as part of TACIT (Han & Baldwin, 2011; Pennington, Socher, & Manning, 2014).

Developing new plugins

One of the main reasons for developing TACIT was to build an architecture that could keep up and evolve with the rapid improvements in computational linguistics and natural-language processing. To achieve this goal, we (1) made the tool open-source, so that users can easily change the internal operations of the program, and (2) built it using the Eclipse platform (www.eclipse.org/eclipse/), which provides an intuitive environment for building new plugins, therefore allowing the tool's capabilities to continue to expand as new developments and data needs arise. TACIT provides a public API for getting access to features that are already form part of TACIT. To develop a new plugin, researchers will need to follow the standard Eclipse Plugin Environment development procedure, discussed in www.eclipse.org/pde. We also provide instructions and examples in the TACIT wiki (http://cssl.usc.edu/wiki/index.php/TACIT) of how researchers can integrate their plugins into the program. Furthermore, these new plugins can be uploaded to the TACIT server and made available for other researchers.

Discussion

The digital age has increased the interconnectivity of individuals around the world, and our daily lives have become awash with communication. As computers have provided researchers with the ability to employ innovative research designs and reach broader populations, so too have they provided almost limitless access to naturally occurring language. Automated tools combined with theoretically driven hypotheses can help leverage this new source of data to expand our understanding of how language use reflects who we are and how we interact. TACIT was developed to integrate the steps and tools for text analysis research into one easily navigable tool, and its open-source, plugin architecture facilitates its continued growth with changing technology and scientific needs.

Similar to the early days of many other types of open-source software, TACIT currently has a lot of shortcomings. First, any software that abstracts several different types of functionality under the same architecture is prone to bugs during its developmental period; a simple change in one API could cause significant problems in other plugins. Second, although TACIT has been implemented as a plugin architecture, the infrastructure for developing new plugins is currently not completely user-friendly. Due to this issue, and also to the fact that new NLP software is written on different platforms, porting of these new approaches to TACIT might not yet be as smooth as we want them to be. Third, even though TACIT has attracted a relatively large number of users in a short period of time, the TACIT community has yet to form and become active. Without support from the community, the continued growth of any open-source software would be slow.

Our hope is that TACIT can facilitate the integration and use of advancements in computational linguistics in psychological research, and by doing so can help researchers make use of the ever-growing documents of our social discourse in ways that have previously not been possible.

Author note We thank Jesse Graham, Kenji Sagae, and Jeremy Frimer for their comments.

References

Abell, M. (2014). SAS Text Miner [Software]. Available from www.sas.com/en_us/software/analytics/text-miner.html

Akthar, F., & Hahne, C. (2012). *RapidMiner 5 operator reference*. Cambridge: RapidMiner.

Andrzejewski, D., & Zhu, X. (2009). Latent Dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing* (pp. 43–48). Stroudsburg: Association for Computational Linguistics.

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

Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2010). WEKA—Experiences with a Java open-source project. *Journal of Machine Learning Research, 11,* 2533–2541.

Breiman, L. (2001). Random forests. *Machine Learning, 45,* 5–32.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent System Technology*, *2*, 27:1–27:27.

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.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20,* 273–297.

De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media. In *ICWSM 2013*. Boston, MA: Association for the Advancement of Artificial Intelligence.

Deerwester, S., Dumais, S., Landauer, T., & Furnas, G. (1990). Indexing by latent semantic analysis. *JASIS*. Retrieved from www.cob.unt.edu/itds/faculty/evangelopoulos/dsci5910/LSA_Deerwester1990.pdf

Dehghani, M., Johnson, K. M., Hoover, J., Sagi, E., Garten, J., Parmar, N. J., . . . Graham, J. (2016). Purity homophily in social networks. *Journal of Experimental Psychology: General*, *145*, 366–375. doi:10.1037/xge0000139

Dehghani, M., Sagae, K., Sachdeva, S., & Gratch, J. (2014). Analyzing political rhetoric in conservative and liberal weblogs related to the construction of the "Ground Zero Mosque.". *Journal of Information Technology & Politics, 11,* 1–14. doi:10.1080/19331681.2013.826613

Duggan, M., Ellison, N. B., Lampe, C., Lenhart, A., & Madden, M. (2015). *Social media update 2014*. Washington: Pew Research Center.

Eichstaedt, J. C., Schwartz, H. A., Kern, M. L., Park, G., Labarthe, D. R., Merchant, R. M., . . . Seligman, M. E. P. (2015). Psychological language on Twitter predicts county-level heart disease mortality. *Psychological Science*, *26*, 159–169. doi:10.1177/0956797614557867

Frank, E., Hall, M., Reutemann, P., & Trigg, L. (2006). *Weka 3: Data mining software in Java*. Hamilton: University of Waikato.

Frimer, J. A., Aquino, K., Gebauer, J. E., Zhu, L. L., & Oakes, H. (2015). A decline in prosocial language helps explain public disapproval of the US Congress. *Proceedings of the National Academy of Sciences, 112,* 6591–6594.

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,* 1029–1046. doi:10.1037/a0015141

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

Han, B., & Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 368–378). Stroudsburg, PA, USA: Association for Computational Linguistics.

Hart, R. P. (1984). *Verbal style and the presidency: A computer-based analysis*. New York: Academic Press.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science, 349,* 261–266. doi:10.1126/science.aaa8685

Corp, I. B. M. (2011). *SPSS Modeler 16 algorithms guide*. Armonk: Author.

Ireland, M. E., Slatcher, R. B., Eastwick, P. W., Scissors, L. E., Finkel, E. J., & Pennebaker, J. W. (2011). Language style matching predicts relationship initiation and stability. *Psychological Science, 22,* 39–44.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32,* 241–254.

Klimt, B., & Yang, Y. (2004). The Enron Corpus: A new dataset for email classification research. In *Machine learning: ECML 2004* (pp. 217–226). Berlin, Germany: Springer.

Kyle, K., & Crossley, S. A. (2015). Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly, 49,* 757–786.

Le, Q. V., & Mikolov, T. (2014). *Distributed representations of sentences and documents*. arXiv [cs.CL]. Retrieved from http://arxiv.org/abs/1405.4053

Leech, G., Garside, R., & Atwell, E. S. (1983). The automatic grammatical tagging of the LOB Corpus. *International Computer Archive of Modern and Medieval English Journal, 7,* 13–33.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In Machine learning: ECML-98 (Lecture Notes in Computer Science, Vol. 1398, pp. 4–15). New York, NY: Springer. doi:10.1007/BFb0026666

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics, 15,* 474–496. doi:10.1075/ijcl.15.4.02lu

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability Vol 1* (pp. 281–297). Berkeley: University of California Press.

Mander, J. (2015). Global Web Index Social Summary Q1 2015. London: Global Web Index.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60). Stroudsburg, PA, USA: Association for Computational Linguistics.

Math, C. (2014). The Apache Commons Mathematics Library. Retrieved from commons.apache.org/proper/commons-math/, September 8, 2013.

McCallum, A. K. (2002). MALLET: A machine learning for language toolkit [Software]. Retrieved from http://mallet.cs.umass.edu

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge: Cambridge University Press.

Mehl, M. R., Gosling, S. D., & Pennebaker, J. W. (2006). Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *Journal of Personality and Social Psychology, 90,* 862–877. doi:10.1037/0022-3514.90.5.862

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., & Lin, C.-C. (2015). Package "e1071." Retrieved from https://cran.r-project.org/web/packages/e1071/index.html

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26* (pp. 3111–3119). Cambridge: MIT Press.

Mukherjee, A., & Liu, B. (2010). Improving gender classification of blog authors. In *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing* (pp. 207–217). Stroudsburg, PA, USA: Association for Computational Linguistics.

Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin, 29,* 665–675.

Nguyen, M.-T., & Lim, E.-P. (2014). On predicting religion labels in microblogging networks. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1211–1214). New York, NY, USA: ACM.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems* (pp. 159–168). New York, NY, USA: ACM.

Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). *Linguistic inquiry and word count: LIWC [Computer software]*. Austin: Liwc.net.

Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). *Linguistic inquiry and word count: LIWC 2001*. Mahwah: Erlbaum.

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology, 77,* 1296–1312. doi:10.1037/0022-3514.77.6.1296

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP 2014), 12,* 1532–1543.

Porter, M. F. (2001). Snowball: A language for stemming algorithms. snowball.tartarus.org. Retrieved from http://snowball.tartarus.org/texts/introduction.html

Porter, M. F. (2006). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems, 40,* 211–218.

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

Stirman, S. W., & Pennebaker, J. W. (2001). Word use in the poetry of suicidal and nonsuicidal poets. *Psychosomatic Medicine, 63,* 517–522.

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

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,* 24–54.

T. M. I. Project. (2015). How millennials get news: Inside the habits of America's first digital generation. Retrieved November 8, 2015, from www.mediainsight.org/PDFs/Millennials/Millennials%20Report%20FINAL.pdf

Van Mieghem, P. (2011). Human psychology of common appraisal: The Reddit score. *IEEE Transactions on Multimedia, 13,* 1404–1406.

Wojcik, S. P., Hovasapian, A., Graham, J., Motyl, M., & Ditto, P. H. (2015). Conservatives report, but liberals display, greater happiness. *Science, 347,* 1243–1246. doi:10.1126/science.1260817

Yang, H., Zhuang, T., & Zong, C. (2015). Domain adaptation for syntactic and semantic dependency parsing using deep belief networks. *Transactions of the Association for Computational Linguistics, 3,* 271–282.

Yu, B., Kaufmann, S., & Diermeier, D. (2008). Classifying party affiliation from political speech. *Journal of Information Technology and Politics, 5,* 33–48.