Arab Spring: From News Data to Forecasting

Kathleen M. Carley, Kenneth Joseph, David Filonuk, Geoffrey Morgan, Jürgen Pfeffer

Abstract

Agent-based simulation models are an important methodology for explaining social behavior and forecasting social change. However, a major drawback to using such models is that they are difficult to instantiate for specific cases and so are rarely re-used. We describe a text-mining network analytic approach for rapidly instantiating a model for predicting the tendency toward revolution and violence based on social and cultural characteristics of a large collection of actors. We illustrate our approach using an agent-based dynamic-network framework, Construct, and newspaper data for the sixteen countries associated with the Arab Spring. We assess the overall accuracy of the base model across independent runs for twenty different months during the Arab Spring.

Introduction

In large communities, the social structure and the culture tend to co-evolve leading to social change, and sometimes to revolution. As scientists, we ask what factors contribute to the onset of revolution and the drift of communities to violence. Addressing such a question requires reasoning about a massive, complex dynamic system. Reasoning about such systems is difficult, in part because we are unable as humans to keep all the information and variables in our heads and to think through all the inherent non-linearities. Technology, however, can be crafted to support human reasoning in this regard.

Computer simulation models, in particular agent-based models and system dynamic models, are a critical methodology for supporting human reasoning about complex systems (Epstein and Axtell 1996; Gilbert 2007). These modeling technologies are especially valuable for reasoning about complex socio-cultural systems. A “feature” of these technologies is that as the models become more realistic, they become more capable both diagnostically and predictively. Creating such a realistic model generally requires instantiating it using a wide variety of data for the case in question. Such data often needs to be extracted from a wide range of sources, put into the same time frame, fused and otherwise massaged before it can be used to instantiate the model. Thus, the process of using data to instantiate the model often takes longer than it does to build and debug the model. Due to the huge person-time cost of instantiating a model, modelers often build shortcuts into the simulation process to make special use of special data. The result is methodologies, including toolkits for instantiation and entire simulation engines, which are basically usable only for the context for which they have been instantiated.

To increase the usability of computer simulation technology to study socio-cultural phenomena, we see two methodological pieces that are necessary. First, a validated, stable simulation framework is necessary that provides general, basic mechanisms of human social behavior. On top of this framework, models for specific questions (e.g. state revolution) can then be rapidly pieced together by domain experts without worry of the validity of the underlying simulation engine. A unified simulation framework also provides a stable interface for input that is not amenable to unique processes or data hard-coded into the simulation engine itself. The second methodological tool is a means for automatically, or at least more automatically, instantiating simulation models with empirically grounded data. Reducing the challenge of building a simulation, providing a uniform mechanism for data entry and the automatic instantiation of models make for simulations that are easier to reuse and results that are easier to validate. See Carley et al. (2012) for a discussion of validation techniques.

The focus of the present work is to provide an overview of recent developments along these two methodological fronts. We give an approach that reduces the instantiation challenge for computer simulation models by extracting the data for the model from online information sources using automated and semi-automated processes. This data is then fed into the dynamic-network modeling framework Construct (Carley 1991; Carley 1990), and in particular a multi-level extension of Construct. Construct accepts data in a uniform format, specifically in the form of a series of networks, which allows the framework to be used independently of the data at hand. Finally, on top of this framework, we build a model specific to the context at hand.

Our work expands on previous efforts (Lanham, Morgan, and Carley 2014) by introducing technology that scales better to large numbers of agents and that is more rigorously automated. These efforts define the beginning of a new set of methodologies that has the potential to make simulation models of large social systems easier to build, reuse, and validate. Two advances have made our approach possible. First, advances in text mining that support the extraction of multi-mode, multi-link networks, i.e., meta-networks, from texts have made it possible to automatically extract and make sense of relevant data. Second, the evolution of agent-based models into agent-based dynamic-network models have changed the basic representation of the data needed for instantiation from rules to networks, specifically meta-networks (Carley 2002). Thus, our work is particularly valuable for agent-based dynamic-network models where the actors of interest are discussed in documents that can be text-mined.

Herein we illustrate our advances by applying them to the Arab Spring, a series of revolutionary activities, some of which were violent, that began in December 2010, proceeded for the ensuring year, and in some cases are still on going. We pull data from over 400,000 news articles from Lexis-Nexis from July 2010 to February of 2012 and use unsupervised and semi-supervised machine learning techniques to draw out the meta-networks necessary for instantiation of an agent-based dynamic network model. The focus of the model is on the dispersion of two beliefs – one centered on the need for revolution and one centered on the need for violence. The model reflects the idea that there are multiple constituencies, linked in a network, each with a set of knowledge. Some of this knowledge leads agents to view revolution and/or violence positively or negatively. This model is then run using the Construct framework in tandem with empirical data drawn from the newspaper text we study.

While much work has considered the Arab Spring, most has been in the context of how new media affected the revolution (e.g. Hamdy and Gomaa 2012; Papacharissi and de Fatima Oliveira 2012) or in developing a better understanding of specific long-term factors that led to revolution (e.g. education, Campante and Chor 2012). In contrast, we focus on uncovering processes in the short term that may have led to violence and revolution, but also those that led certain nations to retain stability and peace. These efforts are more in line with recent work by Pfeffer and Carley (2012), who take a portion of the data used in the present article and explore the dynamics of different terms during the Arab Spring.

Our model generates predictions over the twenty months of data independently on a set of sixteen countries that have been identified as having the potential to be swept up in the frenzy of the Arab Spring. In essence, each month of the data represents a natural experiment that we can use to assess whether or not the model makes valid predictions about each of the sixteen states. This generates, in essence, twenty re-uses of the basic state revolution model, and is the re-use context we will use in this paper. Results are analyzed on a per month, per country basis and contrasted with what actually happened using an outside data source that gives the number of protests in each country in each month studied. We observe that the model captures high level patterns in the validation data, lending credence to the methods described, but that work is necessary to improve the model to increase accuracy.

The paper continues as follows: in Section II, we provide a methodological overview to give the reader a broad overview of the technical approach taken. Section III details the simulation framework, Construct, used in the present work, and Section IV details the state revolution simulation model we build on top of this simulation framework. Section V details how we move from newspaper data to the data necessary to instantiate the model. Section VI provides results of the simulations and Section VII concludes with limitations and prospects for future work.

Methodological Overview

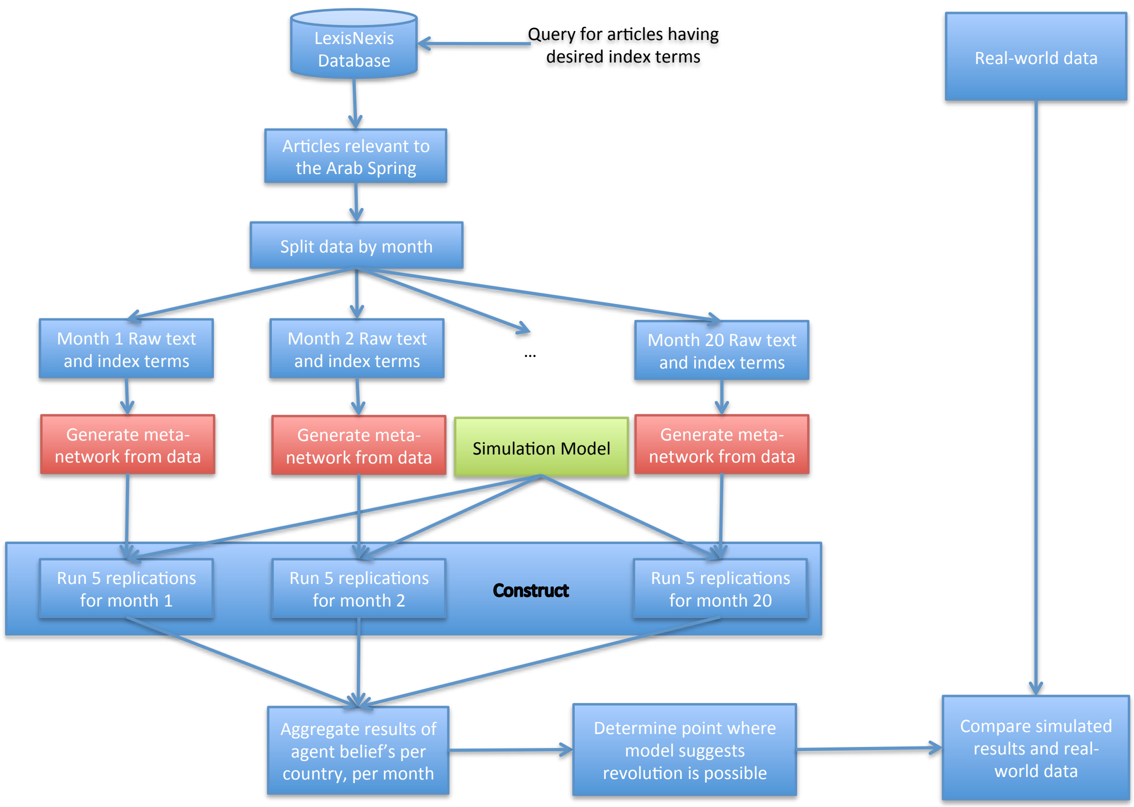

Figure 1

Table 1 - Search terms used to obtain the sample of newspaper articles from Major News Publications from the *LexisNexis* database

Algeria	Bahrain	Egypt	Tunisia	Iran	Yemen
Islamic Republic of Iraq	Jordan	Kuwait	Syria	Lebanon	Western Sahara
Libyan Arab Jamahiriya	Morocco	Oman	Saudi Arabia	Qatar	United Arab Emirates

Figure 1 describes the workflow used in this article. We begin by pulling raw data, composed of newspaper articles from major English-based world news drawn from LexisNexis Academic. LexisNexis uses a proprietary algorithm to associate each article with a set of *index terms*. We query the LexisNexis database for any articles indexed by one of the eighteen terms listed in Table 1 from the period of July 2010 to February 2012. In total, approximately 400,000 articles were collected. Note that while the original data collection used these eighteen keywords, only sixteen countries were used for analysis. This is because we were not able to obtain enough data on two nations, Western Sahara and Qatar, to merit further investigation.

After obtaining the data, we split it up by month. Note that we will hereafter refer to a *set of articles* as the collection of all articles within a single month. From this data, we generate a meta-network that serves as the input for our simulation model. A meta-network is a collection of different nodeclasses that are linked via networks. Thus, our methodology for pulling meta-networks from the LexisNexis data requires us to pull interesting entities (nodes) of different types from the text and uncover relationships between them.

Much work has been done in the area of drawing social structure from text, and the efforts to draw inferences about social networks has increased in recent years. These recent efforts have made a distinct push towards semi-supervised (e.g. Diesner and Carley 2008)and unsupervised (e.g. Eisenstein et al. 2010) learning approaches to extracting social information from text to combat the volume of information that can readily be extracted from the web today. In the present work, our focus is on providing a suite of both semi-supervised and unsupervised techniques that draw out social networks, or more specifically, meta-networks, from text. This approach is grounded in recent efforts showing the power of the meta-network concept in drawing out social structure from text (Diesner and Carley 2008; Diesner, Frantz, and Carley 2005; Diesner and Carley 2004; Lanham, Morgan, and Carley 2014), though the methods are distinctly unique.

After drawing out our meta-network from the text, the Construct framework is passed this data along with the simulation model we build specifically to analyze the Arab Spring. Thus, the meta-network we generate from the text is used to *instantiate* the simulation model. Note that each month, because the simulations are based on stochastic processes, it was necessary to complete replications of each model (for each month). Each of the instantiated models was simulated eight times using slightly different parameterizations of the model, as prior study on the number of replications needed to generate robust results using Construct showed that anything more than five runs were sufficient for result stability (Lee and Carley, n.d.). Thus, this number of replications was deemed sufficient to generate a robust ensemble estimate of the average time to revolution or outbreak of significant violence.

Finally, we use model output to determine the point, if any, at which the model predicts a revolution in a particular nation is likely. While this outcome could be compared to a variety of real-world data sources, we choose to compare decisions made from model output to real-world data on nations involved in the Arab Spring where the government in power was successfully overthrown.

In the following three sections, we detail our simulation framework, Construct, the specific model instantiated here and how we extract the meta-network necessary to instantiate the model. Note that we provide this discussion in the reverse order that these processes are introduced in Figure 1. This will allow us to explicate what data the model and the framework need, and consequently give an overview of what is must be mined from the newspaper articles.

Construct, the simulation framework

Humans tend to hold social relationships with those that they are similar to (Lazarsfeld and Merton 1954; M. McPherson, Lovin, and Cook 2001;Kossinets and Watts 2009). These relationships are bound together by persistent interaction (Licoppe 2004). When humans interact with similar others, they share information and beliefs. This sharing, when moved beyond the context of two individuals in a social relationship into a dynamic social network, creates cascades of information and beliefs that *diffuse* throughout the network (Rogers 2003).

A plethora methods have been put forth to model the diffusion between homophilous actors within a dynamic social network (e.g. Buskens and Yamaguchi 2002; Centola and Macy 2007; Pfeffer and Carley 2013). In many of these models, agents hold a set of "knowledge bits" that spread throughout a network over time. An agent determines whom to interact with (and hence spread information or beliefs to) based on the similarity of his knowledge to the knowledge of his alters, a form of homophily. The process by which knowledge and belief structures co-evolve with the interactions between agents has been defined as *Constructuralism* (Carley 1991).

One weakness of earlier homophily based diffusion models addressed by Constructuralism is that agents in earlier models tended to be locally omniscient- each agent had a perfect perception of the knowledge bits of his alters. In reality, humans work not with a precise notion of the knowledge of others, but rather with a *perception* of it based on previous interactions. Constructuralism addresses this using a mechanism similar to the transactive memory system described by (Wegner 1995). In a transactive memory-based simulation, agents update their perception of the knowledge and beliefs of their alters when interactions occur. This perception is then used to determine whom to interact with based on perceived homophily.

Construct is a turn-based, agent-based dynamic-network framework (Carley 1990; Carley, Martin, and Hirshman 2009) that implements the theory of Constructuralism and has been widely used to examine how ideas diffuse and beliefs change as a function of the underlying social structure in the community. Construct has been validated several times, most recently by Schreiber and Carley (2012). Validated versions of Construct implemented agent cognition as perceived homophily using a transactive-memory based system. However, while transactive memory moves toward a more realistic simulation of the principle of homophily, a purely transactive-memory based model of agent cognition belittles the fact that humans constantly construct their image of both themselves and those around them at higher-order aggregations than the individual. Mead (1925) argued that humans utilize the concept of the *generalized other,* a perception of the knowledge and beliefs of everyone around us based on what we have learned in previous interactions of those around us and ourselves.

Mead's conceptualization of a single generalized other suggests that humans constantly stereotype the knowledge and beliefs of others based on what we can infer or recall about them. With weaker ties, we rely on observable characteristics and things we can infer or recall about another person (e.g. their occupation) to construct a stereotypical view of their knowledge and beliefs[1]. In truth, it is thus our *constructed perceived homophily* that influences our likelihood of interaction with others in a homophily-based diffusion model, rather than either of the previous mechanisms used to model interaction.

The version of Construct used in the present work has been advanced to incorporate a more cognitively plausible and computationally feasible model of the construction of perceived homophily based on the concept of *constructed perceived homophily*. This model was docked to previous, validated versions of Construct by Morgan et al. (n.d.) and is described in more detail theoretically by Joseph et al. (2014). Here, we briefly review the functionality of this tool. To avoid confusion, we will refer to the version of Construct utilized here as *Multi-level Construct (MLC)*.

MLC is based on the notion that agents use two *levels* of familiarity to construct the knowledge of a possible interaction partner. The level of cognition an agent uses to construct the knowledge of a possible interaction partner (alter) is based on the strength of the tie between them. Where the tie is stronger, an agent will have a more precise cognitive representation of his alter's knowledge. More specifically, agents who frequently interact will have an individual-level perception of the knowledge and beliefs of each other, leading to a model of strong ties that is faithful to the original conceptualization of constructuralism and a transactive-memory based scheme. With a weaker tie, however, an agent must construct his alter's knowledge via a process of stereotyping.

The agent constructs what he perceives the alter to know from what he knows of the alter's existence in higher-order social structures. Thus, for example, a revolutionary, Agent A, from Egypt who has rarely interacted with someone from Syria, Agent B, may construct what he expects Agent B to know based on the fact that Agent B is from Syria. In the present work, we place agents into groups based on their beliefs about revolution and violence in addition to their nationality. Thus, Agent A may know that Agent B is from Syria, and that he is associated with a social group that is strongly in favor of revolution.

Agents update their perception of higher-order social structures as they interact with members of them. Thus, upon interaction, Agent B may pass information to Agent A about how to stage a successful protest. Agent A would then ascribe the knowledge of successful protests to Agent B and to the higher-order social structures that Agent B belongs to. Agent A would then be more likely interact in the future with other agents belonging to the same higher-order social structures as

[1] See (Greenwald and Banaji 1995; Hilton and von Hippel 1996) for reviews of the plethora of social psychology literature addressing stereotyping.

Agent B, given that he expects that anyone in the group will share his knowledge of how to stage such a protest.

While by no means a true of human cognition, this segmentation of agent perception into stratified levels that ebb and flow over time is more faithful to how humans construct our perception of those around us. On a more practical level, it also allows us to run significantly larger simulation models. More specifically, a naïve implementation of agent cognition using only transactive memory would require a matrix of size *O(Number of Agents * Number of Agents * Number of Knowledge bits)* to function. A matrix of this size dominates the memory cost of a social simulation, and updating it can have huge effects on time complexity as well. Previous work (Morgan et al., n.d.) shows that MLC has an average space complexity of closer to *O(Number of Agents * Number of Groups * Number of Knowledge Bits)*, which in practice reduces space constraints by an order of magnitude.

Simulation Model

MLC, as a simulation framework, allows for a rich set of data and agent functionality- for details on possible inputs and functions, see (Carley et al. 2012). The data and functionality required for a specific research question are detailed within a simulation model, specified to the framework as an XML document. This model can then be run using the underlying Construct architecture, which implements the theory described above.

With respect to functionality, our state-revolution model is run for 30 turns, suggesting that each turn of the model is roughly equivalent to the duration of a single day. We allow each agent two interactions per model turn, where agents can pass a maximum of two knowledge bits on each interaction. In our model, agents are able to interact with any of the other agents in the simulation. However, agents are seeded to have had previous interactions with others based on information in the raw data (explained below), making them more likely to interact with these individuals. Thus, agents are initially inclined to interact with those that they co-occurred with in the articles in the given month. However, as agents learn the groups of those that they interact with, they will become more likely to interact with those who share similar knowledge and beliefs, leading to a homophily-based diffusion model as the simulation progresses.

Table 1- The necessary nodeclasses and networks to be mined from the raw text

	Belief nodeclass	Agent nodeclass	Agent Groups nodeclass	Knowledge nodeclass
Agents	Agent by Belief network	Agent by Agent network	Agent by Group network	Agent by Knowledge network
Knowledge	Knowledge by Belief network			

With respect to the data needed for instantiation, Table 2 gives a description of the meta-network necessary for our state revolution simulation model. It shows that four types of nodes are necessary- beliefs, agents, agent groups and knowledge. We pull agents and knowledge from the raw text data and infer their relationship to beliefs and agent groups, which we as modelers create. Our model is driven by these latter two nodeclasses, which are higher-order structures. A belief in Construct clusters together a set of knowledge nodes that are relevant to a certain higher order concept. In our state revolution model, we focused on change in the pro-revolution and pro-violence sentiment of the Arab world, and we thus use *violence* and *revolution* as our two beliefs. Beliefs have both positive and negative sentiment- as such, we will have knowledge aligned to both pro and anti sentiments for each of the two beliefs. An agent group clusters together agents that we as modelers believe are associated in some way that others in the social system might form a stereotype about them. In our model, this will be agents who are from the same country and agents that hold similar beliefs.

Table 2 also shows that we require five networks to instantiate the model. The Agent by Agent network details agents that may somehow be related, and thus those that will have a higher likelihood of interaction at the beginning of the simulation. The Agent by Knowledge network specifies the concepts pulled from the text that each agent is associated with, and consequently those they may share with others during the simulation. Both agents and knowledge are also associated with beliefs. The Agent by Belief network specifies an agent's current sentiment for both violence and revolution. The Knowledge by Belief network specifies how different concepts pulled from the text are associated either positively or negatively with the two beliefs. Finally, the agent groups that each agent is in are specified in the Agent by Agent Group Network.

In order to instantiate a model for a particular month, we thus require information on the "who", "what" and "where" of the events occurring in the sixteen countries of interest and relationships between entities in this realm. While this data could have been collected by hand or taken from Subject Matter Experts, the interest of the present work is in a rapid and repeatable process for model instantiation. We thus turned to publically available data and automated methods to pull the information we required for our state revolution model.

Generating the meta-network for instantiation

In this section, we detail how our model is instantiated with the meta-network described above. Before we do so, however, we discuss why the process we use is rather elaborate, as opposed to simply pulling naively from the text. We discuss this in the context of *filtering* and *tuning* the data, and then continue with our discussion of instantiation.

Filtering and Tuning

Practically, the data used to instantiate a simulation model needs to be both filtered and tuned. By filtering we mean the process of selecting the subset of nodes that would be objects in the simulation. By tuning we mean the process of adjusting or adding edges in the network to support analysis. Thus to go from the raw unfiltered data to the networks that would be used to instantiate the simulation model we needed to construct a set of by month meta-networks that included agents, knowledge groups and beliefs.

Why filter? Clearly filtering increases the amount of time it takes to prepare the data to instantiate the model. Thus, keeping filtering to the minimum would be valuable from a rapid construction perspective. On the other hand is the computational cost of the simulation and so the length of time the simulation takes to run. The larger the number of actors and knowledge bits simulated the longer it takes for the simulation to run. Thus, from a purely time-savings perspective there is a tradeoff in time spent filtering versus time spent running the simulator. A small amount of filtering can have large gains in reducing simulation time. The second reason has to do with reducing data bias and increasing the prevalence of relevant data. Recall that at its core the simulation being used examines the diffusion of pro-revolution and pro-violence beliefs within the country in question. To that end, the presence of actors and knowledge not relevant to revolution or violence during the Arab Spring are not only not relevant, they are also distracting from the focus.

The reason such nodes are present in the first place has to do with the nature of the news. Much of the news is associated with events, activities, and actors that have little direct relation to the country in question. An example would be articles about foreign soccer stars who had just played or were about to play a team in the country in question. Another example would be a topic like US gas prices which might occur in an article about how violence in the country in question could impact US gas prices. Thus filtering that removes these extraneous nodes, if it can be done in a rapid, principled and repeatable fashion both enables more rapid analysis and more accurate and focused analysis.

Why tune? Like filtering, tuning increases the time to prepare the data to instantiate the model. Tuning does not speed up model processing. However it is necessary to enable the model to be used at all. We use two types of tuning. First, we use general terms we expect to be associated with our two beliefs to help pull out irrelevant information and agents from the model. This helps overcome news-based biases, and builds on, implicitly, other sources of data. Second, we use the extracted data to infer actor's initial beliefs. We note that the news rarely directly provides input on what stance an actor takes toward the beliefs in a particular model. But to use the model, initial beliefs are needed. Tuning thus makes it possible to infer beliefs. Tuning, if it can be done in a rapid, principled and repeatable fashion both supports data augmentation with secondary data sources and inference of missing data; thus, enabling more detailed and nuanced simulations to be run.

Meta-network creation

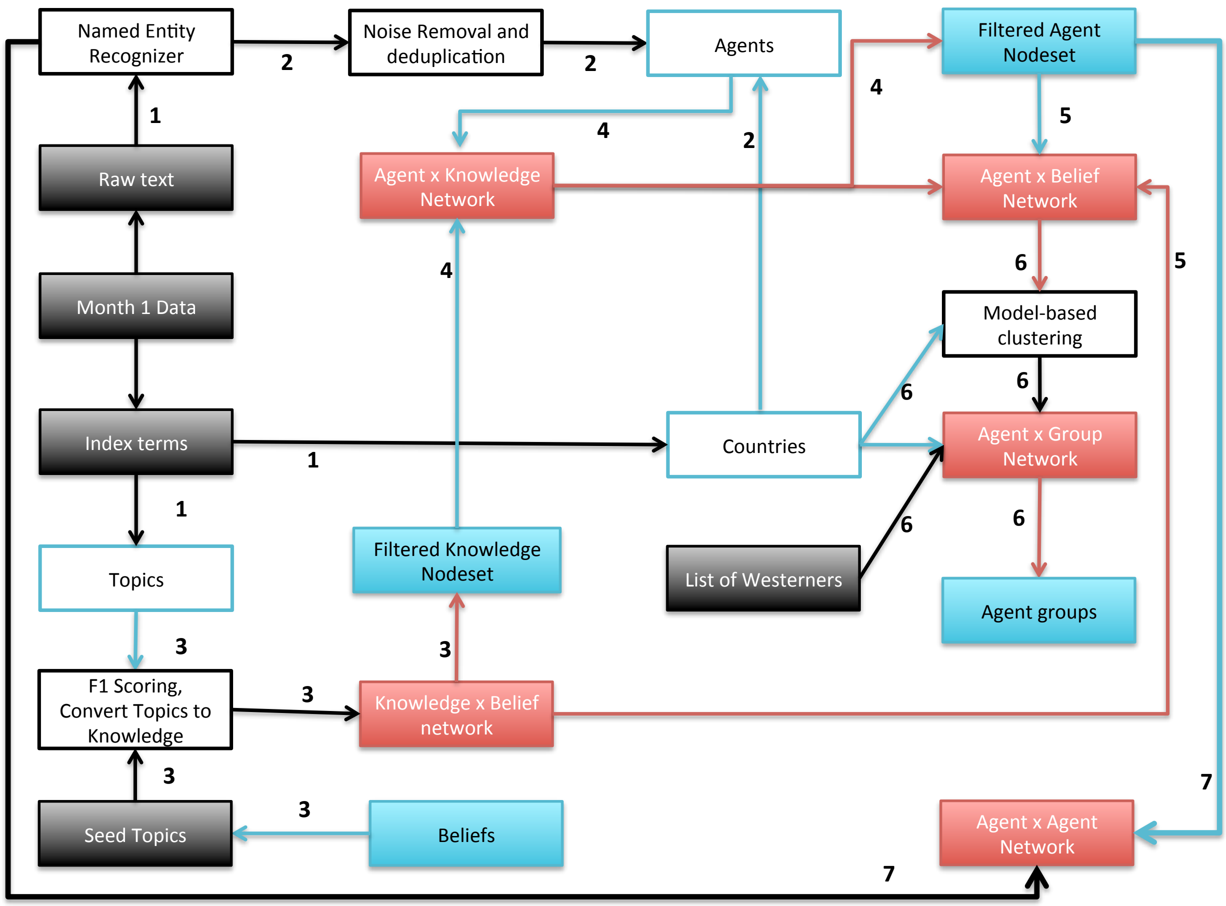

Figure 2

Figure 2 details the seven step process we use to generate a meta-network from a set of newspaper articles. In the figure, there are five different types of boxes. All black boxes represent raw data input to the model, either by the modeler (the seed topics and list of Westerners, as discussed below) or from the LexisNexis articles. White boxes with a black outline represent processes used to manipulate the data. Boxes with light blue outlines represent unfiltered or partial representations of nodeclasses, while the turquoise colored boxes represent the final nodeclasses in the meta-network. Finally, red boxes represent the networks in the meta-network. In addition to the boxes in Figure 2, arrows labeled with step numbers are drawn to indicate the movement of data through the generation mechanism. Below, we detail each of the seven steps represented in the diagram.

Step 1: Obtaining the raw agents, knowledge and countries from the text

We are provided with two sources of information from the set of newspaper articles pulled from LexisNexis- the raw text of the articles and the terms used to index each article. From the indexed terms, we obtain the topics and countries being discussed in any given article. Note that topics are distinct from knowledge, as we will derive the actual knowledge nodeclass used in the simulation from the topics we discover. However, LexisNexis did not include key actors (for example, prime ministers of several of the countries studied) from the Arab Spring in the indexing system. It was therefore necessary to extract the agents discussed in each article directly from the text. In order to do so, we rely on the Stanford Named Entity Recognizer (NER), which uses a Conditional Random Fields approach to pull named entities from a given text (Finkel, Grenager, and Manning 2005). Though the model is trained only on American and English news corpora, we find qualitatively that it provides a reasonable collection of persons involved in the Arab Spring.

Step 2: Filtering of agents

There are four steps in filtering of the agent nodeclass: 1) noise removal, 2) de-duplication, 3) the removal of agents not associated with the countries of interest and 4) the removal of agents not associated with the topics of interest. To reduce noise in the data, we first remove any names discovered by the NER that were of length one (e.g. *Bill*) and any names longer than length five. These names rarely referred to actual people of interest.

De-duplication is completed according to two heuristics. First, we combine into a single actor any names returned by the NER that have a string edit distance of less than four[2]. We find that a distance of three generally suggests alternate spellings of Arabic names by Western journalists, while increasing the threshold any further results in a sharp decrease in the number of agents identified (i.e. too many combinations). Second, we merge into a single agent all names pulled by the NER where one name is *approximately consumed by* another. In order to do so, we take the smaller of the two names (or, where length is equal, either name) and determine the proportion of space-delimited terms in the longer name that are also in the shorter name. If this proportion is greater than or equal to some threshold, then we assume the two names refer to the same actor. Thus, at a threshold of .75 (used here), *Hillary Clinton* and *Hillary Rodham Clinton* would not be merged, while *President Barack Hussein Obama* and *President Barack Obama* would be merged. This heuristic allows us to merge many Arabic names where Western journalists only included "first, middle and last names" and cases where actors' titles are included with their names in some articles and not in others.

Our heuristics fall under a broader category of query correction, for which there exist a variety of both formal (e.g. Li, Duan, and Zhai 2012) and informal[3] techniques. While more formal methodologies show promise in being able to further deduplicate terms produced by the NER, our relatively greedy heuristics provide us

---

[2] See the Wikipedia entry on "Levenshtein distance" for details on what the string-edit distance implies.

[3] E.g. the FuzzyWuzzy python module, https://github.com/seatgeek/fuzzywuzzy, used by companies like StubHub

with a qualitatively reasonable set of actors to simulate in the present work. This is because in many cases noise in the agent nodeclass drowned out by our focus only on higher-order social and knowledge structures.

After noise removal and deduplication, we have a partial set of agents. We wish to further filter this set, however, by retaining only agents associated with one of the sixteen countries of interest and at least one of the topics of interest in a given month. To associate each agent with countries, we use *co-occurrence* information. Co-occurrence information is determined by calculating the number of times a given agent was mentioned in an article indexed by a given country. We associate each agent with a single country, the one that they co-occurred most frequently with. Thus, for example, Hosni Mubarak, if mentioned in ten articles indexed by Egypt and three articles indexed by the United States, would be associated with Egypt. Agents are associated to knowledge through their co-occurrence with the topics this knowledge represents as well. This process, and the final step in filtering the Agent nodeclass, is detailed in Step 4.

Step 3: Generating the Knowledge by Belief Network and the Knowledge nodeclass

Table 2- List of "seed topics" used

Pro Revolution	Anti Revolution	Pro-violence	Anti-violence
CRIMINAL FALSE IMPRISONMENT	EMBASSIES & CONSULATES	BOMBINGS	ARMS CONTROL & DISARMAMENT
ETHNIC GROUPS	FOREIGN RELATIONS	DEATH & DYING	ARTISTS & PERFORMERS
ECONOMIC CRISIS	HEADS OF STATE & GOVERNMENT	FIREARMS	CHARITIES
ELECTION FRAUD	INTERNATIONAL RELATIONS	GENOCIDE	COLLEGES & UNIVERSITIES
HUMAN RIGHTS VIOLATIONS	SOCIAL JUSTICE	SUICIDE BOMBINGS	ENTERTAINMENT & ARTS
POLITICAL CORRUPTION	US PRESIDENTS	TERRORISM	PEACE PROCESS
PROTESTS & DEMONSTRATIONS	INTERGOVERNMENTAL TALKS	TERRORIST ORGANIZATIONS	PEACEKEEPING
RIOTS	TALKS & MEETINGS	TORTURE	SPORTS
MILITARY OPERATIONS	TRADE DEVELOPMENT	CRIMES AGAINST HUMANITY	UNITED NATIONS INSTITUTIONS
BOYCOTTS	TAX TREATIES & AGREEMENTS	WAR & CONFLICT	WEAPONS DECOMMISSIONING

When generating the knowledge by belief network, we must have some indication of the relationship of each topic discussed to a particular valence (positive or negative) of the two beliefs that we use in the model. We then will use the strength of each topic's association with a give belief valence to generate a set of knowledge bits for each topic. In order to determine the valence of topics, we first define a subset of the index terms in the LexisNexis database that are generally associated can be with a valence on the two beliefs of interest. These *seed topics* may not specific to the Arab Spring per se, but are rather concepts for what generally might bring about or result from pro (anti) revolution (violence) sentiments within a population. Table 2 defines the ten *seed topics* associated with positive and negative sentiments along the revolution and violence beliefs that were used in the present work, selected via iterative coding.

Using this set of general topics that relate to particular sentiments of the beliefs in our model, we wish to uncover a set of additional, latent topics that are also associated with our beliefs. In other words, we would like to *expand* the seed topics with terms that are related in some way in our set of articles. This procedure falls under both the domain of sentiment mining (see Pang and Lee 2008, p. 27-28) and automated query expansion (AQE), the process of uncovering terms related to a user's query in order to provide them with more pertinent results. A variety of approaches have been developed for AQE, ranging from highly complex to more rapid approaches (see Carpineto and Romano, 2012, for a review). More complex approaches tend to incorporate contextual information about the document in which terms are placed, for example, the term's position in the document. Because we work only with index terms (terms that may not even be in the document), we choose to utilize a more straightforward methodology that can be carried out based solely on co-occurrence information.

Our method assumes that topics co-occurring frequently with one or more of the terms provided in Table 2 and infrequently without any of these terms will be most related to our two beliefs. We refer to the lists of topics defined in the columns of Table 2 as $\boldsymbol{r}_{+}, \boldsymbol{r}_{-}, \boldsymbol{v}_{+}$and $\boldsymbol{v}_{-}$, respectively from left to right. For each topic in $\boldsymbol{r}_{+} \cup \boldsymbol{r}_{-} \cup \boldsymbol{v}_{+} \cup \boldsymbol{v}_{-}$, we can construct a *topic-article vector (TAV),* $\boldsymbol{t}^{\boldsymbol{topic\_name}}$, that has $|\boldsymbol{A}|$ entries, where $\mathbf{A}$ is the set of all articles in the given month. The *i*th entry of a given TAV, $t_{\boldsymbol{i}}^{\boldsymbol{topic\_name}}$, is a binary value representing whether or not the topic appeared in the *i*th article of the given month.

For each term not in $\boldsymbol{r}_{+} \cup \boldsymbol{r}_{-} \cup \boldsymbol{v}_{+} \cup \boldsymbol{v}_{-}$we can also compute a TAV. We then calculate the similarity of each topic's vector to all topics in Table 3 using a weighted version of the F1 metric similar to the one used by Raina, Ng, and Koller (2006). The F1 metric (equivalent to the Dice coefficient[4]) measures the extent to which a term appears in an article if and only if a second term appears. Thus, the F1 metric as used here can be thought of as the extent to which one topic, represented by the TAV x, occurs only in articles where the other topic, represented by the TAV y, also occurs.

---

[4] Trivially, see http://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/ for a derivation

$$WF1(x, y) = \log(|xy|) * \frac{2\dfrac{|xy|}{|x| * |y|}}{\dfrac{1}{|x|} + \dfrac{1}{|y|}} \qquad (1)$$

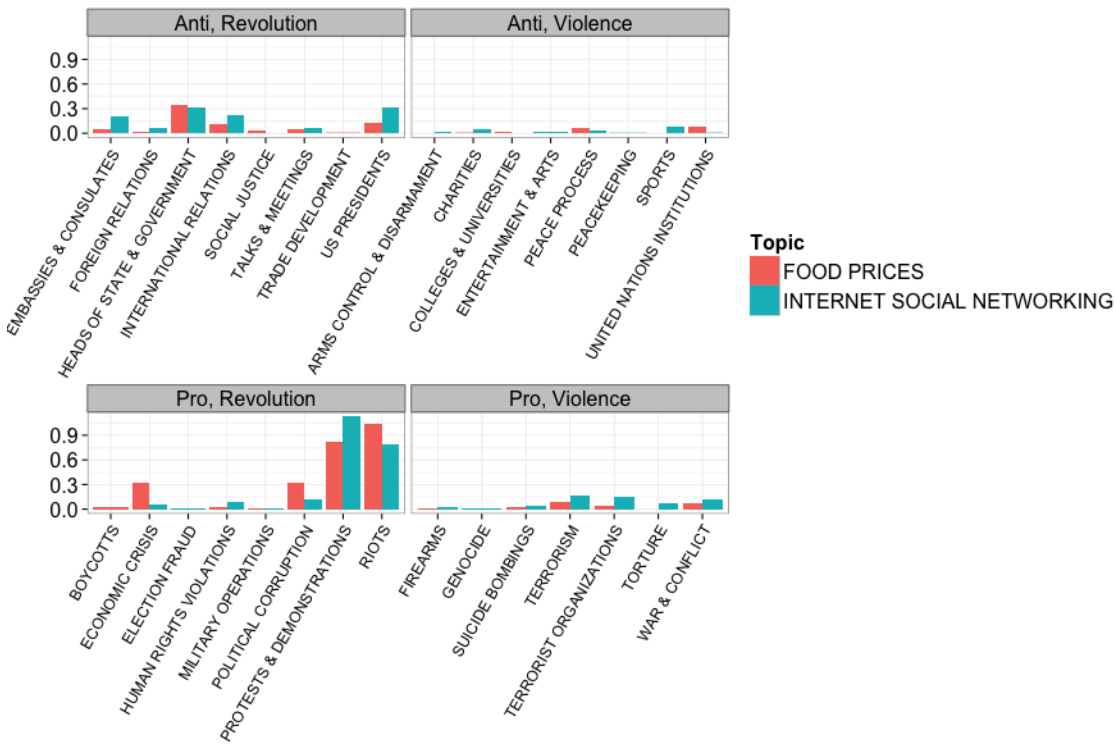

Figure 3

Figure 3 provides a visualization of the weighted F1 similarity score of two topics important to the Egyptian revolution with each of the terms in Table 2 for articles written in January 2011, the month violence broke out. On the y-axis, we observe the outcome of the similarity scores for *food prices* and *internet social networking* to each seed topic. As is clear, food prices are highly associated with positive valences along our revolution belief. A recent article from the Economist[5] suggests that this association is far from trivial, and that "food has played a bigger role in the upheavals than most people realise". Similarly, as previous work has suggested (e.g. Papacharissi and de Fatima Oliveira 2012), social networking online played an important role in the onset of revolution in Egypt. These two examples give anecdotal evidence that our approach provides a useful mechanism to quickly pull relevant topics from newspaper articles.

Having a metric to associate each topic in the data to each term in Table 2, the sentiment of each topic can then be defined via some combination of its WF1 score

[5] http://www.economist.com/node/21550328

with the positive and negative valence terms we specified. We choose here to sum the scores for each term, thus treating each seed topic as an independent indicator of the relevance of any other topic in the text to a given belief. Equation 2 gives the formula used to calculate r, the valence of *food prices* along the revolution belief, for the TAV $\boldsymbol{t}^{\boldsymbol{food}}$.

$$r = \sum_{topic \in \boldsymbol{r}_+} WF1\left(\boldsymbol{t}^{\boldsymbol{topic}}, \boldsymbol{t}^{\boldsymbol{food}}\right) - \sum_{topic \in \boldsymbol{r}_-} WF1\left(\boldsymbol{t}^{\boldsymbol{topic}}, \boldsymbol{t}^{\boldsymbol{food}}\right) \tag{2}$$

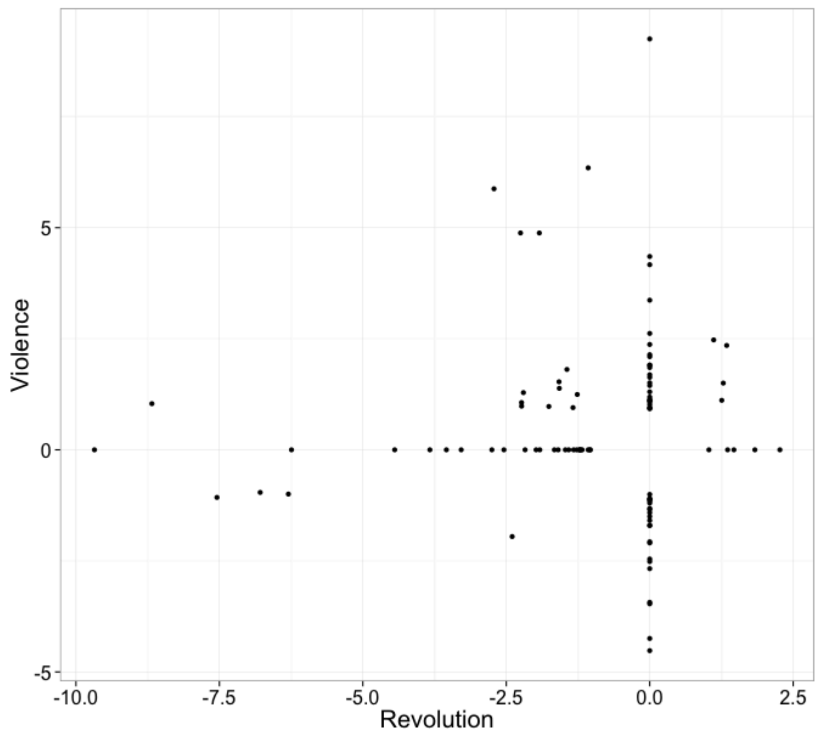

Figure 4

Unsurprisingly, we find that most topics align close to zero along both beliefs. In order to perform filtering, we removed all topics inside of two standard deviations from the mean valence of each belief. Importantly, topics that have a high valence on one belief and not another belief are set to zero on the opposing belief. Figure 4 gives a display of the distribution of valences of all topics used in the model for January 2011, showing a gap near zero on both axes where uninteresting topics were removed or moved to zero.

Having associated topics to beliefs, we now must determine how to move from the abstraction of a topic to the concept of a knowledge bit that is central to our simulation model. While a one-to-one mapping from topic to knowledge bit would be a straightforward solution, we opt for a slightly more complex model to accoutd for the fact that some topics have significantly stronger valence than others. To move from topics to knowledge bits, we allow extra knowledge bits for a given topic based on a logarithmic scaling of the similarity weights. Thus, while in most cases a topic will be represented by a single knowledge bit, some topics, such as *internet social networking*, that have a disproportionately high valence would be represented with multiple knowledge nodes, each of which has a valence of +1 in favor of revolution. This relationship creates the knowledge by belief network.

Step 4: Generating the Agent by Knowledge network and finalizing the Agent nodeclass

The actor by knowledge network actually involves two networks. The first network determines the number of knowledge bits for a particular topic that an agent knows. Where a topic has only a single knowledge bit, an agent will be connected to it in the actor by knowledge network if the two co-occur in any article in the given month. If the topic has more than one associated bit, the agent will only be connected to a larger number of bits if he co-occurs with the topic that number of times. Thus, an agent co-occurring with *protests and demonstrations* three times would be given three of relevant knowledge bits, while an agent co-occurring with the term five or more times would obtain five of the knowledge bits associated with this topic (if five such bits existed). Given the heavy-tailed distribution of co-occurrences between actors and relevant topics in general, this heuristic provides a model suitable for instantiation.

The second actor by knowledge network utilized concerns the likelihood that an agent would transmit any given knowledge bit to an alter on a given interaction. In order to obtain a probability distribution across knowledge bits, we first create a distribution across topics for each agent based on co-occurrences. We rescale agent's associations with all topics to sum to one, giving a probability of transmitting any given topic. An agent will then transmit any given knowledge bit within a particular topic with equal likelihood. Thus, if an agent had a 50% chance of transmitting information about protests and demonstrations on any given turn and knew two of these facts, they would have 25% chance of transmitting either of these bits during interaction.

Step 5: Generating the Agent by Belief network

To create groups of agents according to their beliefs, we first must create a representation of agents' beliefs. This is done by summing the valences of all topics an actor co-occurred with for each belief. Mathematically, we can represent the creation of the Agent by Belief network as the matrix multiplication $I(\boldsymbol{AT})*\boldsymbol{TB}$. The term I($\mathbf{AT}$) represents the binarized form of the agent by topic network, where the index $\mathbf{AT}_{\mathbf{a,t}}$ is 1 if Agent a co-occurred in any article with Topic t, and is zero otherwise. The term $\mathbf{TB}$ simply represents the topic by belief network, where the value $\mathbf{TB}_{\mathbf{t,b}}$ is found via Equation 2.

Step 6: Generating Agent groups and the Agent by Group Network

We consider three types of agent groups in the model. First, we define a *Westerner* group, which is used to define agents that were associated with one of the sixteen Arab Spring countries in Step 1 but who are known to be actors from Western nations. In order to do so, we defines partial names, e.g. "Bush" and "Obama", that were matched against each Agent pulled out from the named entity recognizer. Any name containing the terms provided were placed into the "Westerner" group. These agents were not included in either of the other two grouping methodologies described below.

The second type of groups was based on country. Having aligned each agent with a country in Step 1, these groups are trivial to create. Finally, we create groups based on belief homophily at the global and per-country levels. From the Agent by Belief network obtained in Step 5, we have a two-dimensional representation of agents, which can be considered a latent social space (McPherson and Ranger-Moore 1991). Krivitsky et al. (2009) suggest that an appropriate technique for clustering actors in a latent social space is some form of model-based clustering. Here, we use a Gaussian mixture model to find clusters of agents in the latent belief space.

All clustering was done using the mclust package (Fraley et al. 2012) in R (R Core Team, 2012) with a covariance matrix allowing for variable volume, shape and orientation of the clusters. We determine the optimal number of clusters by taking the model with the optimal BIC, which gives a stricter penalty for "adding" another cluster than AIC or other similar best-fit statistics. However, we only consider a maximum of twenty clusters due to the computational costs associated with attempting groupings at higher levels. Thus, the number of agent groups is variable, based on a clustering of the agents that best fits the data.

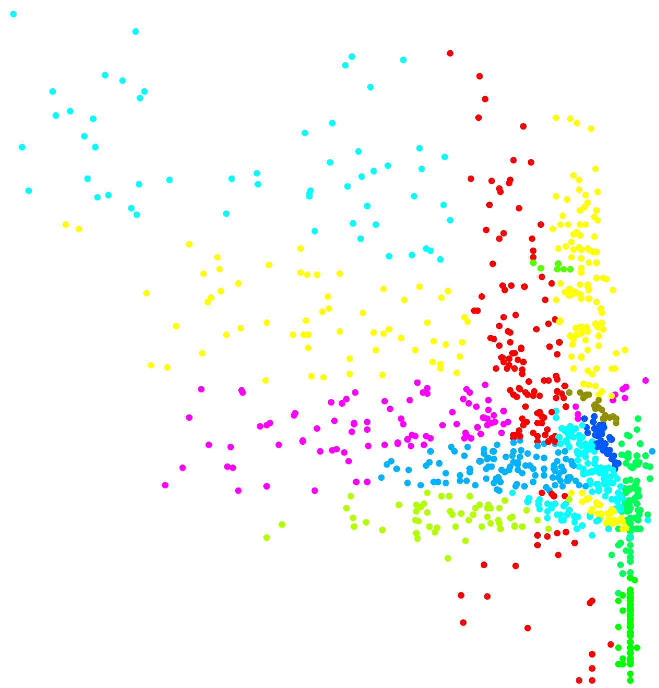

Figure 5

Figure 5 shows the best clustering for January 2011 across all agents in the simulation associated with the country Egypt. Agents are aligned by their revolution belief (the x-axis) and their violence belief (the y-axis). The visualization was created using the network analysis tool ORA (Carley et al. 2012). In the figure, the color of the underlying nodes represents their association with a particular cluster. The origin can be determined by observing the large clustering of agents near the bottom left of the figure. From the smaller variance of the clusters near the origin, we see that most agents are relatively neutral along both beliefs. However, we note that many of the agents were aligned heavily with anti-revolution. These entities were nearly all government officials.

Step 7: Generating the Agent by Agent Network

The final step is to create the Agent by Agent network. This step must be completed last, as our final agent nodeset is not determined until we generate both the agent by topic network and the agent by country network (a subset of the Agent by Group network). Once we obtain the final set of agents, the Agent by Agent network is uncovered simply using co-occurrence data. These relationships are taken to be highly noisy, and are thus used only to seed agents with possible initial interaction partners.

Outcome Description

Overall, we simulate 20 months of data. Eight replications are performed for each month of data, each with slightly different parameterizations of MLC. Parameter differences were motivated by previous work (Joseph, Morgan, and Carley, n.d.), however, we find that there are no interesting differences across parameterizations and hence average across all runs for analytical purposes. With eight replications of each month this is a total of 160 simulation runs, each of which simulates between 7000 and 13000 agents. As output, we first collected the agent by belief network from the last time period for each run and subtract the initial agent by belief network created in Step 5. We then sum the resulting matrix across all agents associated with each of the sixteen countries we study here, giving us the *change in belief* of the agents associated with each country from the beginning to the end of the simulation. Note that the Agent by Belief matrix given at the end of the simulation is formulated and output by Construct -for further details on how Construct computes this network, we refer the reader to (Lanham, Morgan, and Carley 2014).

Because our focus is on initial change points that might indicate a revolution will occur in a particular country, we must now translate the change in beliefs into a binary predictor indication that revolution is possible in a particular country. In order to do so, we leverage our expectation that revolution and violence beliefs in a country on the verge of revolution would be 1) noticeably different than the values of previous months for the same country and 2) noticeably different than the values of the same month for other countries.

Because distributions of beliefs across countries could not be assumed normal, we adopt a non-parametric approach. To determine whether or not a country's revolution and/or violence beliefs were noticeably different than previous months, we first compute the inter-quartile range for each belief of the current and previous two months of data. We refer to these two ranges collectively as the *intra-country ranges*. If a country's belief value for a particular month is outside of the respective intra-country range, it is said to have noticeably changed. Similarly, to determine whether or not a country's revolution and/or violence belief is noticeably different from all other countries in that month, we compute the inter-quartile range of each belief for all countries in that month. We refer to this statistic as the *inter-country range*.

If the changes in a country's violence and revolution beliefs are both outside of its respective *intra-country range* and *inter-country range* in a given month, we determine that revolution is likely. For the purposes of predicting revolution, we use the first such month as our binary predictor of revolution. Thus, we obtain as a final outcome, for each country, zero or one month in which the model predicts revolution has become likely.

Results

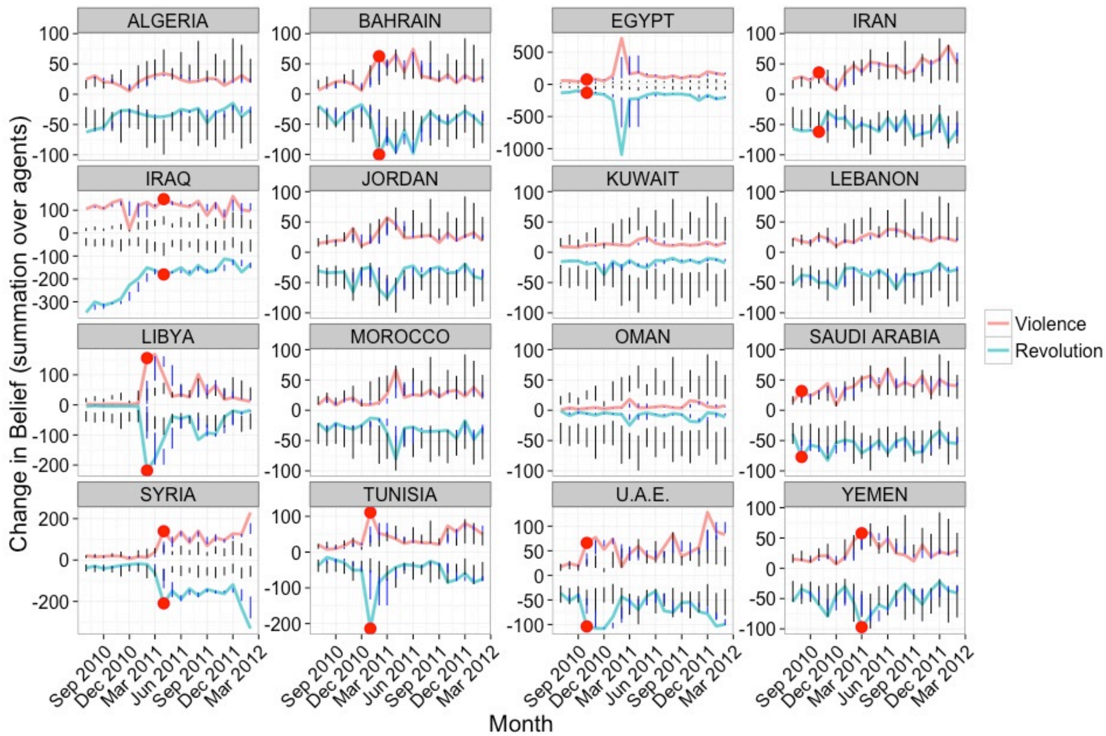

Figure 6

Figure 6 shows the mean change (across the eight replications) in the revolution and violence beliefs for each country and each month. The turquoise line represents change in the revolution belief and the red line represents violence. Vertical black lines at each month represent the *inter-country ranges* and blue lines represent the *intra-country ranges*. Red dots indicate the month in which the model predicts that a revolution is likely. Note that the magnitude of the scale for each country is unique in order to show variation on a month-by-month level for each nation.

Before discussing the relation of model predictions to real-world findings, obvious features of the results are discussed. First, Figure 6 shows that the magnitude of change for the two beliefs closely mirrored each other. This suggests that agents who were frequently mentioned in the context of one belief were often mentioned in the context of the other as well. However, while violence tended to increase when moving outside of a baseline range, revolution sentiment tended to *decrease*. This contradicts our a priori belief that revolution sentiment would increase when protests and revolution occurred in the Arab world. Instead, results and qualitative analysis of the data suggest that when English-speaking journalists discussed the Arab Spring, the focus was on what was being done to construct solutions to the violence that was occurring. These steps often involved actions related to the anti-revolution seed topics listed in Table 2.

Because of this, we adopted our mechanism for predicting revolution to make predictions based on the absolute value of beliefs (and the respective inter-quartile ranges). Thus, revolution was indicated when the revolution belief was noticeably negative and the violence belief was noticeably positive.

Table 3

Country	Model Prediction	Date of Government Overthrow
ALGERIA	None	None
JORDAN	None	None
KUWAIT	None	None
LEBANON	None	None
MOROCCO	None	None
OMAN	None	None
LIBYA	2/11	8/11
EGYPT	10/10	2/11
TUNISIA	1/11	1/11
BAHRAIN	2/11	None (Revolution squashed around 3/11)
SYRIA	4/11	None (Civil War began around 3/11)
YEMEN	3/11	1/11
IRAN	10/10	None
IRAQ	4/11	None
SAUDI ARABIA	8/10	None
U.A.E.	10/10	None

Table 3 shows, for each country, the month and year that the reigning government was overthrown by revolution (or None if this never occurred) and the month our model predicted revolution first became likely. The model correctly raises no sign of revolution in Algeria, Jordan, Kuwait, Lebanon, Morocco and Oman, nations only tangentially associated with the Arab Spring. The model also correctly predicts that revolution would occur in Libya, Egypt, Tunisia and Yemen, the four nations where governments were overthrown during the period of study. Importantly, however, the model does so with varying levels of temporal accuracy. As is clear, model predictions of revolution proceeded actual overthrow by six months, four months and zero months in Libya, Egypt and Tunisia respectively. In Yemen, the model first predicted a revolution would occur two months *after* the government was overthrown, a prediction we consider to be incorrect.

The model also predicts revolution in six nations where governments were not overthrown. In two of these nations, revolution indeed occurred but, as of the writing of this article, has not been successful in overthrowing the government in place. In Bahrain, though the government was not overthrown, significant attempts at revolution were made before being crushed by the reigning regime[6]. In Syria, though the government has not yet been overthrown, a civil war continues that began in late March of 2011. Thus, though model output is here compared to successful revolutions, these two nations represent model output that correctly predicted the rise of revolution.

Finally, the model predicts that revolution was likely in four nations -- Iran, Iraq, Saudi Arabia and the United Arab Emirates – that have seen little coverage as nations where revolution is likely to occur. All four of these countries represent nations with strong diplomatic functions in the region, and thus were the foci of journalists covering the broader impacts of the Arab Spring on the region. Furthermore, the ongoing conflicts between Iraq, Iran and the United States during the timeperiod of study provided additional coverage of these nations not related to the Arab Spring. Consequently, we observe that the model could be improved in the future by learning to differentiate the context in which the topics in Table 2 are covered.

Conclusion

We have presented an approach that enables a simulation model to be instantiated in a semi-automated fashion. The core advantage is this enables model reuse, and supports improved validation. We illustrated this approach using the Arab Spring. We created a state level revolution model using the Construct framework, that we instantiated and ran across twenty different months. The suite of simulations from start to finish the model took less than a week to run (or approx. 6 hours per simulated month start to finish), albeit on powerful hardware[7]. At the technological level our results indicate the value of this semi-automated approach to instantiating an agent-based dynamic network.

On the strict prediction task of determining when a successful revolution would occur in a country before it actually occurred, our model achieved 75% recall at the expense of 30% precision. On the looser scale of correctly predicting significant revolutionary activities in a country, the model does much better, achieving 100% recall and 60% precision. At the theory level, our state-revolution model thus appears to be useful for predicting revolutionary activity.

Model accuracy could very likely be improved via model tuning, in particular the modification of parameters to MLC and changes to the seed topics to provide more complete coverage. This issue of coverage is critical, as in other work we found that immediately prior to the initial revolutionary event the complexity of the topics

[6] See, e.g., http://www.guardian.co.uk/world/interactive/2011/mar/22/middle-east-protest-interactive-timeline.

[7] All replications for a single month were run in parallel using 8 cores of a 60 core machine with a 250 GB SSD drive and 120 GB of RAM.

(number of topics and their interconnectivity in a topic network) and the number of actors of interest actually increases (Pfeffer and Carley 2012). Thus, we suggest that the model could be further improved by a more extensive extraction of data from the news articles, and accounting for the inherent non-linearity due to topic interconnectivity.

The model could also be improved via future, less heuristically based deduplication approaches, as the wholly unsupervised approach we take to entity recognition is likely to provide a moderate level of false de-duplications (i.e. combining two people into one agent) and duplicates. Heuristics for the mechanism we use to obtain a binary predictor could also be tuned. Finally, in the assessment, we kept each month independent in order to demonstrate the reusability of the approach. However, updating each month in a Bayesian fashion using the priors from the month before seems like an appropriate methodology to construct more accurate predictions. Though some model tuning did occur over the course of model development, the focus of the present work was the process by which the model was instantiated and utilized to make predictions. We therefore leave steps taken to improve model accuracy for future work.

Essentially, the limitations of this model theoretically, may be overcome by improving the semi-auto-instantiation process. We purposely took a very high level rapid approach to instantiating the model. Our goal was, in part, to see how accurate the model could be using data that could be rapidly extracted for all countries. We expect further refinements of the instantiation process will lead to improved accuracy in the results from the model. This bodes well for the field of simulation as a real policy tool as it paves the way for making models re-usable through reduced effort in instantiating the model for different contexts.

References

Buskens, Vincent, and Kazuo Yamaguchi. 2002. "A New Model for Information Diffusion in Heterogeneous Social Networks." *Sociological Methodology* 29 (1) (December 17): 281–325. doi:10.1111/0081-1750.00067.

Campante, Filipe R., and Davin Chor. 2012. "Why Was the Arab World Poised for Revolution? Schooling, Economic Opportunities, and the Arab Spring." *The Journal of Economic Perspectives* 26 (2): 167–187.

Carley, K. M. 2002. "Smart Agents and Organizations of the Future." *The Handbook of New Media* 12: 206–220.

Carley, Kathleen. 1990. "Coordinating for Success: Trading Information Redundancy for Task Simplicity." *Institute for Software Research* (January 1). http://repository.cmu.edu/isr/44.

———. 1991. "A Theory of Group Stability." *American Sociological Review* 56 (3) (June 1): 331–354. doi:10.2307/2096108.

Carley, Kathleen M, David T Filonuk, Kenny Joseph, Michael Kowalchuck, Michael J Lanham, and Geoffrey P Morgan. 2012. “Construct User Guide.”

Carley, Kathleen M, Michael K Martin, and Brian R Hirshman. 2009. “The Etiology of Social Change.” *Topics in Cognitive Science* 1 (4) (June 26): 621–650. doi:10.1111/j.1756-8765.2009.01037.x.

Carley, Kathleen M, Jürgen Pfeffer, Jeff Reminga, Jon Storrick, and Dave Columbus. 2012. “ORA User’s Guide 2012”. DTIC Document.

Carley, Kathleen M., Geoffrey Morgan, Michael Lanham, and Jürgen Pfeffer. 2012. “Multi-Modeling and Socio-cultural Complexity: Reuse and Validation.” *Advances in Design for Cross-cultural Activities* 2: 128.

Carpineto, Claudio, and Giovanni Romano. 2012. “A Survey of Automatic Query Expansion in Information Retrieval.” *ACM Comput. Surv.* 44 (1) (January): 1:1–1:50. doi:10.1145/2071389.2071390.

Centola, Damon, and Michael Macy. 2007. “Complex Contagions and the Weakness of Long Ties.” *American Journal of Sociology* 113 (3) (November 1): 702–734. doi:10.1086/521848.

Diesner, Jana, and Kathleen M. Carley. 2004. “Using Network Text Analysis to Detect the Organizational Structure of Covert Networks.” In *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.5240&rep=rep1&type=pdf.

———. 2008. “Conditional Random Fields for Entity Extraction and Ontological Text Coding.” *Computational and Mathematical Organization Theory* 14 (3): 248–262.

Diesner, Jana, Terrill L. Frantz, and Kathleen M. Carley. 2005. “Communication Networks from the Enron Email Corpus ‘It’s Always About the People. Enron Is No Different’.” *Comput. Math. Organ. Theory* 11 (3) (October): 201–228. doi:10.1007/s10588-005-5377-0.

Eisenstein, Jacob, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2010. “A Latent Variable Model for Geographic Lexical Variation.” In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1277–1287. EMNLP ’10. Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1870658.1870782.

Epstein, Joshua M., and Robert Axtell. 1996. *Growing Artificial Societies [electronic Resource]: Social Science from the Bottom Up*. Brookings Institution Press. http://books.google.com/books?hl=en&lr=&id=xXvelSs2caQC&oi=fnd&pg=PA1&dq=Growing+artificial+societies&ots=_fO3_EVKsx&sig=7jEKS8Xxq2SyY32qHGNhpncVZ2s.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling.” In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 363–370. ACL ’05. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1219840.1219885. http://dx.doi.org/10.3115/1219840.1219885.

Fraley, Chris, Adrian E. Raftery, T. Brendan Murphy, and Luca Scrucca. 2012. "Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation". Technical Report. http://cran.r-project.org/web/packages/mclust/vignettes/mclust.pdf.

Gilbert, Nigel. 2007. *Agent-Based Models*. Thousand Oaks, CA: Sage Publications Inc.

Greenwald, Anthony G., and Mahzarin R. Banaji. 1995. "Implicit Social Cognition: Attitudes, Self-esteem, and Stereotypes." *Psychological Review* 102 (1): 4–27. doi:10.1037/0033-295X.102.1.4.

Hamdy, Naila, and Ehab H. Gomaa. 2012. "Framing the Egyptian Uprising in Arabic Language Newspapers and Social Media." *Journal of Communication* 62 (2): 195–211. doi:10.1111/j.1460-2466.2012.01637.x.

Hilton, James L., and William von Hippel. 1996. "Stereotypes." *Annual Review of Psychology* 47 (1): 237–271. doi:10.1146/annurev.psych.47.1.237.

Joseph, Kenneth, Geoffrey P Morgan, and Kathleen M Carley. n.d. "On the Coevolution of Stereotype, Culture and Social Relationships." *In Submission*

Kossinets, Gueorgi, and Duncan J. Watts. 2009. "Origins of Homophily in an Evolving Social Network1." *American Journal of Sociology* 115 (2) (September): 405–450. doi:10.1086/599247.

Krivitsky, Pavel N., Mark S. Handcock, Adrian E. Raftery, and Peter D. Hoff. 2009. "Representing Degree Distributions, Clustering, and Homophily in Social Networks with Latent Cluster Random Effects Models." *Social Networks* 31 (3): 204–213.

Lanham, Michael J., Geoffrey P. Morgan, and Kathleen M. Carley. 2014. "Social Network Modeling and Agent-Based Simulation in Support of Crisis De-escalation." *IEEE Transactions on Human-Machine Systems* forthcoming. http://www.andrew.cmu.edu/user/mlanham/pubs/Lanham2012_IEEE_SMC_Final_Draft.pdf.

Lazarsfeld, PF, and RK Merton. 1954. "Friendship as a Social Process: A Substantive and Methodological Analysis." In *Freedom and Control in Modern Society*, 18–66. Van Nostrand.

Li, Yanen, Huizhong Duan, and ChengXiang Zhai. 2012. "A Generalized Hidden Markov Model with Discriminative Training for Query Spelling Correction." In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 611–620. http://dl.acm.org/citation.cfm?id=2348365.

Licoppe, Christian. 2004. "Connected Presence: The Emergence of a New Repertoire for Managing Social Relationships in a Changing Communication Technoscape." *Environment and Planning D: Society and Space* 22: 135–156.

McPherson, J. Miller, and James R. Ranger-Moore. 1991. "Evolution on a Dancing Landscape: Organizations and Networks in Dynamic Blau Space." *Social Forces* 70 (1) (September 1): 19–42. doi:10.1093/sf/70.1.19.

McPherson, M., L. Lovin, and J. Cook. 2001. "Birds of a Feather: Homophily in Social Networks." *Annual Review of Sociology* (1): 415–444. doi:citeulike-article-id:3022278.

Mead, George Herbert. 1925. “The Genesis of the Self and Social Control.” *International Journal of Ethics* 35 (3) (April 1): 251–277. doi:10.2307/2377274.

Morgan, Geoffrey P, Kenneth Joseph, and K. M. Carley. “Multi-level Construct.” *In Submission*.

Pang, Bo, and Lillian Lee. 2008. “Opinion Mining and Sentiment Analysis.” *Foundations and Trends in Information Retrieval* 2 (1-2): 1–135.

Papacharissi, Zizi, and Maria de Fatima Oliveira. 2012. “Affective News and Networked Publics: The Rhythms of News Storytelling on #Egypt.” *Journal of Communication* 62 (2): 266–282. doi:10.1111/j.1460-2466.2012.01630.x.

Pfeffer, Jürgen, and Kathleen M Carley. 2013. “The Importance of Local Clusters for the Diffusion of Opinions and Beliefs in Interpersonal Communication Networks.” *International Journal of Innovation and Technology Management* 10 (5).

Pfeffer, Jürgen, and Kathleen M. Carley. 2012. “Rapid Modeling and Analyzing Networks Extracted from Pre-structured News Articles.” *Computational and Mathematical Organization Theory* 18 (3): 280–299.

R Core Team. 2012. *R: A Language and Environment for Statistical Computing*. Vienna, Austria. http://www.R-project.org/.

Raina, Rajat, Andrew Y. Ng, and Daphne Koller. 2006. “Constructing Informative Priors Using Transfer Learning.” In *Proceedings of the 23rd International Conference on Machine Learning*, 713–720. http://dl.acm.org/citation.cfm?id=1143934.

Rogers, Everett M. 2003. *Diffusion of Innovations, 5th Edition*. Simon and Schuster.

Schreiber, C., and K. M. Carley. 2012. “Validating Agent Interactions in Construct Against Empirical Communication Networks Using the Calibrated Grounding Technique.” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* PP (99): 1 –9. doi:10.1109/TSMCA.2012.2192104.

Wegner, D. M. 1995. “A Computer Network Model of Human Transactive Memory.” *Social Cognition* 13 (3): 319–339.